

Karoo Array Telescope Site Shielding: Laboratory, Computational and Multi-copter Studies

by

Hardie Pienaar



*Dissertation presented for the degree of Doctor of
Philosophy in Engineering in the Faculty of Engineering at
Stellenbosch University*

Supervisors: Prof. H. C. Reader
Prof. D. B. Davidson

December 2015

Declaration

By submitting this dissertation electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: 2015/9/1

Copyright © 2015 Stellenbosch University
All rights reserved.

Abstract

Karoo Array Telescope Site Shielding: Laboratory, Computational and Multi-copter Studies

H. Pienaar

*Department of Electrical and Electronic Engineering,
University of Stellenbosch,
Private Bag X1, Matieland 7602, South Africa.*

Dissertation: PhD (E&E)

December 2015

The Northern Cape in South Africa has been chosen to host the Square Kilometre Array (SKA) due to the area's overall radio quietness. As part of the supporting systems for the Karoo Array Telescope (KAT), a processing building has been constructed on the site. With the Karoo Array Processing Building (KAPB) now in place, investigating the radio frequency interference (RFI) of the building has become a high priority. If successful, understanding of RFI propagation on-site will shape policies and contribute to the sustainability of on-site radio quietness. This dissertation focuses on understanding the shielding and propagation characteristics of both the KAPB building, as well as a man-made soil berm. On-site measurements, scale models and computational models will be used to investigate the local electromagnetic environment. Additionally, a Multi-copter vehicle is developed to support on-site measurement campaigns. Using data measured on-site it was possible to develop empirical models for local shielding estimations. It was found that the shielding performance of the berm was primarily affected by diffraction. Also, the developed computational model makes it possible to investigate alternative terrestrial structures. The work done in this dissertation will permit off-site analysis of propagation over terrestrial structures. Moreover, the development of a Multi-copter measurement platform creates more efficient metrology. Finally, empirical models are designed so that shielding budgets can be calculated for noise sources to the nearest receivers.

Uittreksel

Karoo Array Teleskoop Omgewing Beskerming: Laboratorium, Rekenaar Modelling en Multi-kopter Studies

H. Pienaar

*Departement Elektriese en Elektroniese Ingenieurswese,
Universiteit van Stellenbosch,
Privaatsak X1, Matieland 7602, Suid Afrika.*

Proefskrif: PhD (E&E)

Desember 2015

Die Noord Kaap in Suid Afrika is gekies om 'n gedeelte van die Square Kilometre Array (SKA) te huisves. Die besluit is gedeeltelik gemaak as gevolg van die skoon radio omgewing in die Noord Kaap. 'n Ondergrondse gebou, genaamd die Karoo Array Processing Building (KAPB), is gebou as deel van die ondersteunende infrastrukture van die Karoo Array Telescope (KAT). Met die oprigting van die Karoo Array Processing Building (KAPB), het dit belangrik geraak om radio frekwensie steuring (RFI) te ondersoek. Indien die studie suksesvol is, sal die begrip van RFI voortplanting in die omgewing help om beleid te vorm ten opsigte van die volhoubaarheid van die skoon RFI omgewing. Hierdie proefskrif fokus op die verstaan van beskermings vermoë van beide die KAPB gebou en die mensgemaakte keerwal asook plaaslike radio frekwensie voortplantings karakteristieke. Metings geneem in hierdie area, skaalmodelle sowel as numeriese elektromagnetiese modellering (CEM) is gebruik om die plaaslike elektromagnetiese omgewing te ondersoek. Verder is 'n Multi-kopter platvorm ontwikkel vir die ondersteuning van metings veldtogte. Deur gebruik te maak van gemete data wat geneem is in die area, kon empiriese modelle ontwikkel word wat gebruik is om die plaaslike RFI beskerming te beraam. Daar is gevind dat die beskerming van die heuwel hoofsaaklik geaffekteer word deur die diffraksie daaroor. 'n CEM model is ook ontwikkel om die ondersoek van ander heuwelagtige strukture moontlik te maak. Die werk in hierdie proefskrif sal dit moontlik maak om analise, weg van die Karoo omgewing te doen. Addisioneel, het die ontwikkeling van 'n Multi-kopter platvorm dit moontlik gemaak om metings vinniger te doen. Die uiteinde is dat empiriese modelle dit

moontlik maak om 'n beskermings vermoë begroting uit te werk vir 'n geraas bron tot die naaste MeerKAT ontvanger.

Acknowledgements

I would like to express my sincere gratitude to the following people:

My family who supported and inspired me.

Anneke Stofberg for her dedicated support through all, especially the hard, phases of this project.

Prof H. C. Reader for his wholehearted guidance through the project as well as his passionate teachings in EMC. His impromptu lessons in language will shape my English for years to come.

Prof D. B. Davidson for his valuable guidance and inputs during the last year of my PhD.

Braam Otto and Paul van der Merwe from MESA for their support in my Multi-copter and dielectric measurement efforts. They kept their belief, even in times when things started flying away.

Simon Norval for arranging and making most of the measurements in this dissertation possible.

Wessel Croukamp for his support and advice in all things mechanical.

Johan Frank for giving me the confidence and guidance to pilot a Multi-copter for the first time.

The past and present members of the EMRIn group: Matthew Groch, Antheun Botha, Nardus Mathysen, Stefan Combrink, Joely Andriambeloson for a place to share thoughts and ideas as well as their contribution to all the interesting moments in room E212.

This project was funded by SKA South Africa and CHPC.



Figure 1: Photo of our group during the last Karoo measurement campaign discussed in this work. From top left: Stephan Combrink, Simon Norval, Carel van der Merwe, Paul van der Merwe, Nardus Mathyssen, Braam Otto. From bottom left: Hardie Pienaar, Antheun Botha, Gideon Wiid.

Contents

Declaration	i
Abstract	ii
Uittreksel	iii
Acknowledgements	v
Contents	vii
List of Figures	x
List of Tables	xix
Nomenclature	xxii
1 Introduction	1
1.1 The SKA and RF propagation studies	1
1.2 Dissertation objectives	3
1.3 Contributions	3
1.4 Layout of the dissertation	4
2 Background/Literature Study	6
2.1 SKA South Africa, the KAPB and RFI	6
2.2 Computational modelling, propagation, and knife-edge diffractions	7
2.3 Existing Multi-copter-based metrology vehicles	18
2.4 Dielectric measurements	21
2.5 Conclusion	26
3 Dielectric Measurement	27
3.1 Introduction	27
3.2 Waveguide measurement	27
3.3 Free-space measurement	33
3.4 Coaxial measurement	38

3.5	Dielectric fit	39
3.6	Conclusion	41
4	Computational Modelling and Scale Measurements	42
4.1	Introduction	42
4.2	Measurement of the scale model	43
4.3	Computational modelling using an FDTD code	46
4.4	Quasi-infinite ground plane	48
4.5	Computational modelling using a MoM code	52
4.6	Full berm simulation	54
4.7	Conclusion	64
5	Multi-copter Metrology Development	66
5.1	Introduction	66
5.2	Vehicle design	67
5.3	Receiver development	69
5.4	Antenna integration	75
5.5	Antenna verification	79
5.6	Electromagnetic signature and EMC recommendations	81
5.7	Measurement extraction	83
5.8	Conclusion	89
6	Propagation and Shielding Measurements	91
6.1	Introduction	91
6.2	Testing the KAPB shielding effectiveness	91
6.3	Ground loss	98
6.4	Characterisation of the berm	100
6.5	Site-base shielding budget	105
6.6	Comparing measurements to the computational model	109
6.7	Verification of the simplified propagation model	111
6.8	Conclusion	114
7	Conclusions and Recommendations	115
	Appendices	118
A	Waveguide Measurements	119
A.1	Calibration standards	119
A.2	Calibration and repeatability validation	120
A.3	Improved stepwise Nicolson-Ross-Weir Extraction	121
B	Free-space Dielectric Measurements	125
B.1	Fields in the focal region of horn antennas	125
B.2	Offline calibration of measured S-Parameters	126
B.3	Calculated error coefficients	127

CONTENTS

ix

B.4	Offline calibration and extraction code	128
B.5	Root combinations	129
B.6	Parameter fitting	133
C	Coaxial Measurements	136
C.1	Material parameter extraction	136
C.2	Morris Rimbi code	140
D	Quasi-infinite Plate Optimisation Program	142
E	Multi-copter Software, Antennas and Specification Sheets	151
E.1	RFM22B daughter board	151
E.2	3D-Antenna Patterns	153
E.3	Specifications sheets	156
E.4	Raspberry-Pi logging software	158
F	Aerial Real-time Transient Analyser Development	167
F.1	In-lab power-line sparking measurement	167
F.2	Real-time transient analyser design	168
G	Antenna Beam Scanning	210
H	Multicopter Measurements	213
	Bibliography	216

List of Figures

1	Photo of our group during the last Karoo measurement campaign discussed in this work. From top left: Stephan Combrink, Simon Norval, Carel van der Merwe, Paul van der Merwe, Nardus Mathysen, Braam Otto. From bottom left: Hardie Pienaar, Antheun Botha, Gideon Wiid.	vi
1.1	Perspective from the top of the 13 m high berm facing towards Losberg. The KAPB and assembly shed can be seen to the left. The core of MeerKAT is located on the opposite side of Losberg. Photo by Dr P. G. Wiid.	2
2.1	Photo showing three of the seven KAT-7 telescope dishes. KAT-7 is the precursor to MeerKAT.	7
2.2	Diagram showing how the geometry of the wire example has been approximated into linear segments. The left side is the original problem where currents could exist in arbitrary orientations on the wire surface. On the right, currents are constrained onto the axial direction of the wire.	9
2.3	Schematic of the transmission line example. The transmission line itself has been discretised into lumped-element components. . . .	12
2.4	Diagram illustrating a 2D Yee grid for the transmission line example. Here the staggering of the voltage and current points can be seen. The variables involved in calculating a single voltage point are also demonstrated.	13
2.5	Illustration of Huygens principal. A source emits a wavefront into a wedge obstacle. The third wavefront demonstrates how energy is diffracted into the shadowed region by using Huygens principle. .	17
2.6	Photo of LS of SA using their Multi-copter platform for a measurement. Receiver and computer equipment are mounted on the bottom of the vehicle with the antenna secured to a fixture between two of the motor arms.	19
2.7	Photo of the IEIIT Group Multi-copter during an array scanning measurement. Dipole antenna and tone generator can be seen mounted on its underside.	19

2.8	Photo of a Multi-copter being used at the University of Nijmegen. The antenna, in this case, has been placed away from the vehicle. Photo was taken from Krause [48].	20
2.9	An illustration of a material subject to an applied E-field. After [52].	22
2.10	Parallel plate illustration. After [52].	22
2.11	Free-space measurement system.	24
2.12	X-band waveguide measurement system during a through calibration. Material holders and calibration sections can be seen in the background.	25
2.13	Coaxial probe system.	25
3.1	Photographs of the S-Band and X-Band waveguide measurement sets including calibration standards and sample holders.	28
3.2	S-Band samples with Teflon already fitted into the sample holder. Teflon and Perspex will be used for reference measurements. . . .	29
3.3	S-Band waveguide time-gated S-parameters.	31
3.4	X-Band waveguide time-gated S-parameters.	31
3.5	S-Band and X-Band permittivity extracted with the improved NRW algorithm.	32
3.6	Offline calibration using a load, short and offset short. At this point the short is measured at $z = 0$ with an absorbing foam block, also the load, behind it.	34
3.7	Model of S-parameter systematic errors.	35
3.8	Offline calibrated measurement of the foam on the free-space system.	36
3.9	Calibrated S-parameter measurements of the foam with reference planes shifted.	36
3.10	Root combination minimum error plot at 11.6 GHz. This plot solves the left-hand side of Eq. 3.3.5.	37
3.11	Free-space extracted parameters of the foam. A slight ripple is present in the extraction due to calibration simplifications.	38
3.12	Dielectric data over all of the three measurement campaigns. . . .	40
3.13	Fitted polynomial and log trend.	40
4.1	Diagram of the metallic plate used for scale measurements. The antennae are located at each end of the plate, with the aluminium mounted on a wooden base for extra mechanical support.	44
4.2	Diagram (left) and photo (right) of the fat monopole antenna, mounted on the metallic plate, with its feeding connector on the bottom side of the structure. The antenna was machined out of brass material.	44
4.3	Photo of the scale-model measurement setup inside the anechoic room.	45
4.4	Scale model, without dielectric, measured outside of the anechoic chamber with a 10 ns time-gate.	45

4.5	Antenna feed mesh in CST MWS [®] . Here it can be seen how all of the detailed features have been meshed. Additionally the connector dielectric section has been given extra mesh cells to account for its shorter wavelength.	46
4.6	Computational models constructed in CST MWS [®] . A simulation with and without the dielectric was used for the measurement comparison. The dielectric and the non-dielectric measurement contained respectively 227 and 190 million mesh cells.	47
4.7	Both measurement and simulated scenarios, without the dielectric (a), and with the dielectric (b), are compared. A more efficient infinite approximation is also shown in (a).	47
4.8	Illustrating how rays are diffracted on a thin PEC serration with an oblique angle. To illustrate the optimisation case, a 2D version is also shown.	49
4.9	Diffracted ray comparison for an optimised and non-optimised plate, using a simplified GTD approximation. The dot at the top of the plate represents the transmitting antenna while the surrounding square indicates the zone that is optimised to be free of diffractions. In the non-optimised case, red rays indicate rays that have passed through this zone.	50
4.10	Testing of the serrated plate compared to the infinite case using CST MWS [®] . In (a) the S_{21} coupling is compared showing good agreement between the time-gated-serrated measurement and the actual infinite case. The time domain received wave can be seen in (b), showing the direct and reflected waves. TG - Time-gated. . .	51
4.11	Constructed serrated plate with its non-dielectric measurement. Here the serrated time-gated measurement is compared to a full infinite plate CST MWS [®] simulation.	52
4.12	FEKO [®] PEC half-space compared to infinite CST MWS [®] simulation in (b). The two antennas can be seen on both sides of the image, on top of the infinite PEC halfspace.	53
4.13	FEKO [®] mirrored simulation including dielectric. Results here are compared to quasi-infinite measurements.	54
4.14	Full berm simulation with antenna 50 m away. The berm and antenna were both mirrored around the ground plane to make RLGO in FEKO [®] possible.	55
4.15	Simulation of a vertical plane, showing the E-field through the centre of the berm at 260 MHz. The source used in this simulation is the simple resonant monopole on the ground. The solution is not valid inside dielectric volumes. Therefore, the area that contained the berm is left blank.	55
4.16	Fresnel integral as a function of v . This graph was taken from Seybold [22].	56

4.17	Simulated shielding, with a monopole on the ground, 50 m in front of the berm at 260 MHz.	57
4.18	Shielding simulations to a nearby MeerKAT location for the berm at heights of 4, 10 and 40 wavelengths. The border from the line-of-sight to the shadowed regions are indicated by white lines. The KAPB location is roughly outlined by dashed lines in the figures. This marked volume is what is used to calculate figures of merit called ASE and MSE.	58
4.19	Simulated 2D planes using the mirrored berm approximation. The vertically polarised source is located at the position of receiver M60 15 m off of the ground, 5 km away. The planes illustrate the absolute electric field diffracted over and around the obstacle.	59
4.20	The same simulation configuration as in Fig. 4.19 with shed structures added.	60
4.21	Berm shielding effectiveness in KAPB area to MeerKAT receiver M60 at 260 MHz. In this case, a large PEC shed structure has been added, causing diffraction into the shielded area.	60
4.22	Comparison of our PEC, PMC ground cover approximating real-earth using a 2D FDTD model. Both the diffraction (non-grazing angle) and shielding (grazing angle) scenarios are tested.	61
4.23	Diffraction flight path simulations at 50m and 200m.	63
4.24	Horizontal plane measurement made by LS of SA compared to the FEKO [®] model. Measurements directly behind the berm are mostly in the noise floor. However, the propagation around the berm is still informative.	63
4.25	Berm loss-tangent parameter study. The loss-tangent was varied initially by only 20% up and down. Due to the insignificant change, the loss-tangent was further changed by an order of magnitude. This shows how diffraction over the berm converges with an increase in conductivity.	64
5.1	Multi-copter CAD model compared to a photo of its realised counterpart. Plastic parts were 3D-printed while metallic structure parts were obtained from a local hardware store.	67
5.2	Photo and diagram of flight electronics. In (b), the connections between the different electronics can be seen.	68
5.3	Photos of the RFM22B transceiver module. In (b), the transceiver is shown on its daughter board attached to the Raspberry Pi. . .	70
5.4	Software diagram of the logging script running on the Raspberry-Pi. This python script is initialised before each flight and is only terminated after the flight has ended.	71
5.5	The calibration file used during post-processing for a pre-amp receiver pair. This file was created using a known signal-generator as an input while logging the output of the receiver.	71

5.6	System diagram of the real-time transient analyser.	72
5.7	Front-end switchable filter-bank configuration.	73
5.8	Photo of the digital board containing the FPGA, ADC and clock generator.	73
5.9	Photo of the unshielded switch-able filter bank. Filters are implemented using lumped elements.	74
5.10	Photo showing the layout of the dual antenna configuration. In (b), the isotropic antenna pattern formed by an idealised dual-configuration can be seen.	75
5.11	A photo in-flight showing the grounded-copper shield enclosure. The compass and GPS due to the nature of their operation are still placed on the top of the cube. Photo was taken by Dr. P. G. Wiid.	76
5.12	The loading profile of the monopole can be seen in (a). In (b) a picture of the initial simulation model, used in CST MWS [®] , can be seen. One half of the leg here is removed to reveal the antenna with its loaded elements.	77
5.13	Image of the full Multi-copter CEM model. Here the simulated input reflection of the antenna is compared to its measured equivalent.	78
5.14	Realised gain of Multi-copter antennas on a constant phi cut. Cut locations were chosen to align with the deepest nulls of each pattern. During measurement extraction, the data from the antenna with the highest realised gain will be used. Here the difference with increasing frequency can be seen.	78
5.15	Multi-copter mounted on the near-field scanning device. The highlighted (white line) is the active antenna during the measurement.	79
5.16	Measured 2D theta/phi antenna patterns. Data transformed to far-field from a near-field scanning measurement.	80
5.17	Simulated 2D antenna patterns using CST MWS [®]	80
5.18	Measurement of Multi-copter RFI signature using a reverberation chamber. The measurement can be seen in (a) with a photo of the Multi-copter in the chamber in (b).	82
5.19	A simplified schematic of the Multi-copter illustrating the clean-room dirty-room concept. The receiver is shielded from the internally-generated RFI, illustrated in grey.	83
5.20	De-embedding measurements using receiver calibration data and simulated antenna patterns. Deviation between the fully extracted data in (c) is a result of slight simulation deviations together with the uncertainty of the principle incoming signal's direction.	87
6.1	Contemporary photo of the KAPB, dark roof, taken from the top of Losberg. The berm and the assembly shed can be seen on the left and right of the KAPB respectively. Photo was taken by Dr P. G. Wiid.	92

6.2	Diagram illustrating how shielding, S , is calculated around the KAPB, at points x , while transmitting from different rooms, r , inside the building.	93
6.3	Shielding measurement of the KAPB at three discrete frequencies. Ground measurements were done using a hand-held spectrum analyser while measurements at 10 m height were accomplished using the Multi-copter platform. The measurements can be compared to the diagram in (d) for a sense of orientation.	94
6.4	Shielding measurement along the berm-side of the KAPB compared to ground measurements on top of the berm. Vertical lines indicate the start and stop of the KAPB berm-side wall.	95
6.5	Vertical-plane signal strength graph, measured between the KAPB and berm. The diagram in (b) gives an overview of how the measurement was planned.	96
6.6	Vertical shielding measurement, on the berm side of KAPB. This plot shows, at two discrete frequencies, how shielding tends to decrease with height. A diagram showing how the measurement was conducted can be seen in (b).	97
6.7	Minimum shielding vs frequency trend extracted from polar pattern shielding graphs. A square fit was used to approximate the measurements, which can now be used as an analytical tool. . . .	98
6.8	Ground loss measurement compared to the simple ground loss model at two different heights. A free-space prediction can also be seen for comparison. A schematic in (b) gives an overview of the measurement setup.	99
6.9	A diagram of the berm and KAPB environment. Here the location of the antenna can be seen. A region called sampled space was measured to obtain the vertical-plane measurements in Fig. 6.10.	100
6.10	Vertical plane measurements of a noise source transmitting over the berm. At the two discrete frequencies diffraction is seen in the shadowed region of the berm.	101
6.11	A diagram of the diffraction measurements. Here the two measurement locations along with the transmitting position, on the far side of the berm, can be seen. Three discrete frequencies were measured at each location.	102
6.12	Different diffraction model geometries used in the characterisation of the berm. These models range from infinitely-thin PEC knife edges to finitely conducting wedges.	103
6.13	Comparison between real measurement and model configurations seen in Fig 6.12. Measurement was taken at 550 MHz 200 m behind berm.	103
6.14	Comparisons between measurement and calibrated model for berm diffraction. These graphs represent two different positions as well as two different frequencies behind the berm.	104

6.15	Google Earth satellite map showing the location of the KAPB and berm relative to the nearest MeerKAT receiver.	106
6.16	Shielding to nearest MeerKAT receiver (M59) from inside KAPB, outside KAPB and outside KAPB not shielded by the berm. Height for the outside RFI positions were kept at 2 m high 70 m from the foot of the berm. In (b) the maximum allowable signal transmitted, from inside the KAPB, can be seen graphed using (a).	107
6.17	Comparison between the computational models and Multi-copter measurements for a diffraction flight 200 m behind berm at 900 MHz.	110
6.18	Geometry used to calculate when the PEC reflections affect the simulation.	111
6.19	Comparison between model predictions and on-site measured data for the three main scenarios. Dashed and solid lines respectively illustrate the difference between using the two-ray or Egli model for ground propagation.	112
6.20	Received level at M59 and M60 receivers using the measured emissions in the KAPB, translated using the developed empirical models. An upper and lower bound for shielding were calculated using the models from this work. SARAS levels are added to asses RFI severity.	113
A.1	S-Band measurements made over multiple occasions to illustrate repeatability of measurements.	120
A.2	X-Band measurements made over multiple occasions to illustrate repeatability of measurements.	121
A.3	Comparison between typical and improved NRW code. As a result of choosing the incorrect root, the extraction deviates. The point at which this happens is illustrated as the wrapping point in the phase graph.	121
B.1	Amplitude in dB of the radiation pattern in the E-plane along the z-axis for a frequency of 13.5 GHz. Difference between contours are 2dB with a -2dB contour in the middle.	125
B.2	Measured standards used in offline calibration.	126
B.3	Calculated error terms used to calibrate measurements offline. . .	127
B.4	Root combinations found over frequency for minimum error using Eq. 3.3.5.	130
B.5	Root combinations error over frequency.	130
B.6	Root combination minimum error plot at 12.39 GHz. This plot solves the left-hand side of Eq. 3.3.5.	131
B.7	Root combination minimum error plot at 18 GHz. This plot solves the left-hand side of Eq. 3.3.5.	131

C.1	Russouw Probe with its calibration standards. A short on the left and matched load on the right.	137
C.2	Marais Probe with its calibration standards. From left; a short, matched load and a characterised open.	137
C.3	Samples used in coaxial probe measurements. From left; Foam material, Rexolite, Perspex and Teflon.	138
C.4	Extracted foam measurements using Rossouw probe. The influence of the material structure can be seen in the random deviations of the measurements. The dashed line indicates the waveguide measurement for the same band.	139
D.1	In (a), an illustration of variable edges used to build the optimised geometry. The first edge here is already fixed while the second edge is being varied to optimise its angle. Additionally, the discretisation of an edge to calculate diffraction interactions can be seen in (b).	142
D.2	Illustration of diffraction on a serrated plate. A simple non-optimised serration geometry was used, varying length of the vertexes.	143
E.1	Daughter board schematic, drawn using KiCad [87].	152
E.2	PCB design for the RFM22B daughter board. The PCB is responsible for connecting the transceiver module to the Raspberry-Pi. Additionally, an SMA connector for the antenna is supplied.	153
E.3	Multi-copter far-field antenna patterns. These realised-gain patterns give an indication of the field at three discrete frequency points. Continued in E.4.	154
E.4	Multi-copter far-field antenna patterns. These realised-gain patterns give an indication of the field at three discrete frequency points.	155
E.5	Specification sheet for the MT2216 (800 KV) dc-brushless motor [89].	156
E.6	Electrical Specifications of the ZX60-P105LN+ from Mini-Circuits.	157
E.7	Performance Graphs of the ZX60-P105LN+ from Mini-Circuits.	157
F.1	Photo showing the spark-gap on the 12 kV line. The LPDA receiving antenna can be seen as the orange structure directed at the spark-gap.	167
F.2	Sparkling measurement showing the broadband nature of the pulses extending to 400 MHz. A background measurement distinguishes the environment from sparking.	168
F.3	Root page	169
F.4	Clock page	170
F.5	ADC page	171
F.6	FPGA page	172
F.7	Power page	173
F.8	Front Copper	174
F.9	Inner layer 1 Copper	174

F.10 Inner layer 2 Copper	175
F.11 Back Copper	175
F.12 Silk screen	176
F.13 Filterbank PCB Layout	176
F.14 Filterbank schematic	177
F.15 Spectrum plot with 50 Ω load connected to the input.	208
F.16 -12.5 dBm 90 MHz tone applied at input.	209
F.17 -22.5 dBm 90 MHz tone applied at input.	209
G.1 Multi-mode antenna placed in the middle of a large sports field aligned East to West. In this case, the Rat-race coupler was used to drive the antenna in its horizontal dipole mode.	210
G.2 E-plane dipole antenna measurement using Multi-copter compared to laboratory characterised data. Deviation in measurements are caused by the null region de-embedding of on-board antennas. . .	211
H.1 Diffraction measurements at two discrete points behind berm at 260 MHz. These measurements are compared to the calibrated knife-edge model.	213
H.2 Diffraction measurements at two discrete points behind berm at 550 MHz. These measurements are compared to the calibrated knife-edge model.	214
H.3 Diffraction measurements at two discrete points behind berm at 900 MHz. These measurements are compared to the calibrated knife-edge model.	214
H.4 Vertical-shielding measurements for the remaining three sides of the building. The berm side has been discussed in the text. . . .	215

List of Tables

2.1	Brief comparison between the fundamental aspects of the discussed CEM codes.	8
2.2	Summary of different terrestrial propagation mechanisms at various frequencies.	16
2.3	Examples of some near-earth propagation models.	17
2.4	Comparison table between three different dielectric measurement methods used in this dissertation [53].	26
3.1	Overview and comparison of S and X-band waveguide systems. . .	28
3.2	Comparison between known and measured values for Perspex and Teflon. The reference values are both average values over their given frequency bands.	32
3.3	Calibration standard definitions for the free-space measurement system. The measured traces for each standard will be named according to M_1, M_2 and M_3	34
3.4	Expressions for the error coefficient in the model seen in Fig. 3.7.	35
3.5	Lowest error root combinations uniqueness table	38
3.6	Amount of linear test error for each order fit.	39
4.1	Frequencies on the scaled model compared to the full-size berm. .	42
4.2	Dimensions of the characterised foam block.	43
4.3	FOM results in the RoI for the 3D RL-GO and 2D boundary integral models.	59
5.1	A weight estimate of the Multi-copter with all of its systems on-board.	67
5.2	Brushless-dc motor specifications. This data is valid for an MT2216 900 KV motor at 11.1V on an 11x3.7 inch propeller. The rest of the data can be found from the manufacturer's specification sheet in Appendix E.3.	68
6.1	Minimum shielding levels on the berm side of KAPB extracted from Fig. 6.3.	97

*LIST OF TABLES***xx**

A.1	Calibration Standards defined for S-Band Waveguide system in CalKit Manager. Using a short, offset short, thru and sliding load calibration.	119
A.2	Calibration class assignments defined for S-Band Waveguide system in CalKit Manager.	119
A.3	Calibration Standards defined for X-Band Waveguide system in CalKit Manager. Using a short, line and thru calibration.	120
A.4	Calibration class assignments defined for X-Band Waveguide system in CalKit Manager.	120
C.1	Real permittivity of known materials, measured using coaxial probes. Each measurement is repeated three times, each with a new calibration.	138
D.1	Coordinates for one-quarter of the optimised plate.	144
E.1	Fabrication specifications for a 4-layer PCB from OSH Park [88].	151

Nomenclature

Abbreviations

ADC	Analog-to-Digital Converter
AGA	Astronomy Advantage Act
ALMA	Atacama Large Millimetre/sub-millimetre Array
ASE	Average Shielding Effectiveness
CAD	Computer Aided Design
CEM	Computational Electromagnetics
COTS	Commercial Off-The-Shelf
EM	Electromagnetic
EMC	Electromagnetic Compatibility
ESC	Electronic Speed Controller
FEM	Finite Element Method
FDTD	Finite Difference Time Domain
FIFO	First In First Out
FOM	Figure of Merit
FPGA	Field Programmable Gate Array
GO	Geometrical Optics
GTD	Geometrical Theory of Diffraction
GPS	Global Positioning System
HV	High Voltage
KAPB	Karoo Array Processing Building
KAT	Karoo Array Telescope
LNA	Low Noise Amplifier
LOS	Line of Sight
LPDA	Log Periodic Dipole Antenna
MoM	Method of Moments
MSE	Minimum Shielding Effectiveness
MUT	Material Under Test
NRW	Nicolson-Ross-Weir
PEC	Perfect Electric Conductor
PLF	Polarisation Loss Factor
PCB	Printed Circuit Board
PMC	Perfect Magnetic Conductor
PML	Perfect Matching Layer
PO	Physical Optics
RaTTy	Real-time Transient Analyser
RC	Remote Control
RF	Radio Frequency

RFI	Radio Frequency Interference
RoI	Region of Interest
RSSI	Receive Signal Strength Indicator
RL-GO	Ray Launching Geometrical Optics
SARAS	South African Radio Astronomy Service
SBC	Single Board Computer
SE	Shielding Effectiveness
SERDES	Serial/Deserializer
SKA	Square Kilometre Array
SKASA	Square Kilometre Array South Africa
SMA	Sub Miniature version A
SPI	Serial Peripheral Interface
TE	Transverse Electric
TG	Time Gate
TEM	Transverse Electromagnetic
UHF	Ultra High Frequency
UTM	Universal Transverse Mercator
VHDL	VHSIC Hardware Description Language
VHF	Very High Frequency
VLA	Very Large Array
VNA	Vector Network Analyser

Constants

c	299792458	m/s
ϵ_0	8.854×10^{-12}	F/m
μ_0	$4\pi \times 10^{-7}$	H/m

Chapter 1

Introduction

1.1 The SKA and RF propagation studies

The Square Kilometre Array (SKA) will be the world's largest radio telescope upon completion. Currently in its pre-construction phase, implementation is planned to start in 2018 with early science possible in 2020. The project will be the world's largest public science data undertaking with data rates reaching ten times more than current Internet traffic around the globe. The SKA will be completed in two phases, the first of which will contribute to around half of the projected one square kilometre collecting area. During phase two, the collecting area will surpass one square kilometre by some margin. Some of the key science goals are to: test the theory of relativity; discover how the universe with its stars and galaxies formed and evolved; answer questions about dark matter and dark energy; find the origin of cosmic magnetism; continue the search for extraterrestrial life and finally make discoveries that cannot at this point be foreseen.

SKA phase 1 is set to achieve a frequency range of 50 MHz up to 14 GHz which is split between the two host countries South Africa and Australia. Approximately a quarter of a million low-frequency aperture array antennas, covering 50 MHz to 350 MHz, will be built in the Murchison Valley of Australia (SKA1 LOW). The rest of the frequency band, 350 MHz to 14 GHz, will be surveyed using around 200 dishes located in the Karoo desert of South Africa (SKA1 MID).

SKA MID1 currently boasts an estimated baseline of 150 km. Resolution and sensitivity will be respectively four and five times higher than that of the Jansky Very Large Array. Survey speed will see a sixty times improvement over the latter telescope. Precursor telescopes are currently under construction to help us understand the necessary challenges. MeerKAT in South Africa and the MWA[1] and ASKAP[2] in Australia are playing a key role in helping us understand the difficulties ahead. SKA Phase 2 will see the expansion of the array into neighbouring African countries. Similar expansion is planned for



Figure 1.1: Perspective from the top of the 13 m high berm facing towards Losberg. The KAPB and assembly shed can be seen to the left. The core of MeerKAT is located on the opposite side of Losberg. Photo by Dr P. G. Wiid.

the Australian components. Full completion would take around a decade with enabling technologies still needing further refinement [3].

In South Africa's SKA developments, the Karoo has been chosen principally for its pristine Radio Frequency (RF) Environment. With the construction of Karoo Array Processor Building (KAPB), self-generated noise and its propagation into surrounding telescopes become a concern. This investigation seeks to quantify shielding effectiveness to nearby MeerKAT receivers from the KAPB site. Empirically-based propagation and shielding models will allow a stronger basis for qualifying on-site infrastructure in terms of its local radio frequency interference (RFI) emissions. As part of the KAPB construction, excavated earth was used to build a berm adjacent to the building. The benefits of this decision were twofold, lowering the cost of transporting the soil and using it as a shielding mechanism between the KAPB and the nearest MeerKAT receivers. A Multi-copter metrology vehicle was designed to facilitate on-site measurement campaigns. Such a platform enables characterisation of the berm and KAPB (see Fig. 1.1) at different heights. In the case of the berm this allows measurements up to 100 m and thus makes it possible to investigate diffraction over a significant range. Apart from the empirical propagation tools, a verified computational model is created to allow exploration into more complex berm configurations. Such configurations could investigate different berm geometries as well as the addition of other on-site obstacles.

Previous work for on-site shielding includes:

- Characterising the absorption properties of the Karoo soil for cable-shielding studies [4].
- Investigating end-fire patterns from power-line sparking using a Multi-copter [5][6].
- Investigating shielding of soil structures using scale-model measurements [7].
- Time-domain studies using a real-time transient analyser and impulse radiating antenna was used to investigate broadband propagation on the site area [8].

It is important to develop tools to expedite measurements. These on-site efforts are time-consuming and become more logistically involved as MeerKAT comes into operation. Therefore, this work has focussed on using these limited measurement campaigns to build empirically-based propagation models, and a verified computational model, for off-site investigations. During these campaigns, a Multi-copter measurement platform considerably increased the possible investigations.

1.2 Dissertation objectives

The primary objective of this dissertation is to develop propagation models for the Karoo site as part of the campaign to study and minimise the effects of self-generated interference. The initial part of achieving this goal is the establishment of a practical computational model of the on-site berm using scale-model and full-scale measurements. To expedite and enhance on-site measurement capabilities, a Multi-copter measurement platform is designed. This vehicle will be employed for the full-scale propagation measurements as well as the KAPB shielding investigations. Finally, full-scale measurements will be used to compile a set of empirical propagation models that will characterise the severity of on-site RFI sources. The scope of this work will not include investigations into small-scale fading, ionospheric and atmospheric effects, and long distance over-the-horizon propagation. This work will mainly focus on the local shielding-effectiveness characteristics of the KAPB site.

1.3 Contributions

The original contributions made in this dissertation include the following:

1.3.1 Development of a Multi-copter for RFI and propagation studies

To expedite and enhance on-site measurement campaigns, a Multi-copter was designed and deployed specifically for radio-frequency interference and propagation studies. Measurements could be fully de-embedded from the vehicle due to a shielded configuration. This made characterisation of the on-board antennas reliable and facilitated the accurate de-embedding of data. The effect of the complex vehicle on the antenna system is circumvented in most other Multi-copter designs by using a decoupled directional antenna or only focussing on relative measurements while ensuring symmetry around the antenna. Additionally, an integrated dual-antenna configuration forming a quasi-isotropic antenna pattern implemented on the vehicle guaranteed a good degree of sensitivity irrespective of its orientation. The antennas formed an integral part of

the Multi-copter design to the point where they were embedded into its landing structures, making use of the surrounding dielectric material to achieve a lower minimum measurable frequency. The conductive shield simplified the electromagnetic structure of the vehicle to the extent that fully verified simulations were possible. These simulations were used to generate patterns critical for the antenna de-embedding process. All of this contributed to a Multi-copter metrology vehicle with the capacity of doing accurate absolute measurements.

1.3.2 Study of berm shielding characteristics

The shielding characteristics of a real-world berm was studied using a computational model derived from laboratory scale measurements and in-field metrology. The simplified 3D computational model was verified using full-scale diffraction measurements. Using this model, shielding metrics could be derived for the 3D berm case. Previously this was only done for an infinite 2D approximation of the structure. It was found from this study that a berm only becomes effective after its height reaches around 10 wavelengths with an average shielding effectiveness of 10 dB. A minimum shielding level of 15 dB is found when the berm height reaches 40 wavelengths.

1.3.3 Compilation of simplified propagation models based on on-site measurements

A set of simple propagation models for local on-site propagation was compiled to analyse the severity of emission sources on near-lying receivers. This was made possible by using the large scale (100 m vertical) diffraction measurements to calibrate a simplified single knife-edge model for the berm. The shielding performance of the KAPB was characterised using the Multi-copter platform. Finally, ground loss models were tested using horizontal measurements at varying heights.

1.4 Layout of the dissertation

The dissertation will start with a literature review covering the background on topics encountered through the text. This will include information on the Square Kilometre Array South Africa (SKASA), radio frequency interference, computational modelling, propagation, knife-edge modelling, Multi-copters and dielectric measurements.

Chapter 3 will focus on the broadband characterisation of a dielectric block. This chapter will make use of a number of dielectric measurement systems ranging from waveguide to freespace and coaxial techniques. Data from all of these measurements will be used to fit a model that will be extensively used in the scale modelling handled in Chapter 4.

Chapter 4 will make use of scale measurements to build a computational model of a man-made soil berm. In this chapter, both finite difference time domain (FDTD) and method of moments (MoM) codes will be investigated. A MoM/GO model will be used with an approximate PEC ground plane. An effort will be made to test the PEC approximation against real earth using an FDTD code. Finally, shielding effectiveness of the full-scale 3D model of the berm will be studied.

Chapter 5 will shift its focus to the development of a Multi-copter to facilitate on-site measurements. This vehicle will be used for the full-scale on-site measurements highlighted in Chapter 6. This chapter will discuss the design of the Multi-copter with its integrated antennas. This will include the calibration of on-board receivers, verification for antennas and the de-embedding process that will be used to extract the measurements.

In Chapter 6, measurements made by the Multi-copter will be used as a basis for compiling a set of propagation tools. These studies will consist of a shielding characterisation of the KAPB, a study of diffraction over the berm and testing of simple ground loss models over flat-earth. These three studies will be combined to form a set of simplified propagation models to analyse the effect of RFI emissions on-site to the closest MeerKAT receivers.

Chapter 2

Background/Literature Study

2.1 SKA South Africa, the KAPB and RFI

The SKA will be split between remote areas in South Africa and Australia. These locations were chosen primarily for their unprecedented radio quietness. South Africa has already built a 7-dish telescope Karoo Array Telescope (KAT-7) see Fig. 2.1 and is currently in the construction phase of MeerKAT. A total of 16 receivers is planned for the end of 2015 out of the total 64 receivers. Science is expected to start in mid-2017. The core will house 48 of these antennas within a diameter of 1 km. The final planned baseline for the completed project is estimated at 8 km. Each dish is steerable with an effective main reflector size of 13.5 m. A sub-reflector (3.5 m) reflects the beam, in an Offset Gregorian configuration, to an indexer tray housing a set of radio receivers. The total height of the structure is 19.5 m high. Data from the receiver digitisers are sent via an extensive network of optical fibres to correlators inside the Karoo Array Processor Building [9]. MeerKAT is likely to be incorporated into SKA phase one. Examples of other arrays already doing science are the Atacama Large Millimeter/submillimeter Array (ALMA) and the Very Large Array (VLA) [10][11].

As radio telescopes, these receivers are easily affected by other terrestrial radio sources. Also, as these receivers are designed to detect signals reaching back to the apparent origin of the universe, RFI from nearby sources could damage the front-end amplifiers or constrain the possible science. Although the Karoo has been chosen for its radio quietness, the project itself will start to introduce RFI. The emission sources could include equipment such as power distribution, vehicles, personal electronics and air-conditioners. Therefore, throughout the life cycle of the project these sources have to be monitored to maintain the radio quietness.

In the early stages of the SKA bid, South Africa introduced the Astronomy Advantage Act (AGA), which was promulgated in 2007 [12]. The purpose of this act was to ensure the preservation of radio quietness in the Karoo area.



Figure 2.1: Photo showing three of the seven KAT-7 telescope dishes. KAT-7 is the precursor to MeerKAT.

This document led to widespread efforts to regulate industrial RF emissions. Consequently, this created a landscape where not only the SKA but other smaller radio telescopes could function in a low-RFI environment.

From the start of MeerKAT construction in 2013, RFI mitigation has remained a priority. Every possible chance to maintain the natural RFI quietness in that region has been exploited. The site base itself has an underground facility (KAPB) which houses power, processing and generation equipment. This building was placed on the opposing side of a large flat-top hill (Losberg). The unearthed ground from the KAPB was used to build a berm next to the facility with the hope of increasing its shielding. Previous work has been done to establish shielding for underground cabling [4], sparking off of power lines [6] and measuring soil properties for scale models [7]. In parallel with this dissertation, measurements were done using an impulse radiating antenna [8] and real-time transient analyser [13]. These measurements have the capability of characterising a continuous frequency band for shielding studies.

2.2 Computational modelling, propagation, and knife-edge diffractions

2.2.1 Computation modelling [14] [15]

Computational electromagnetics (CEM) is now a mature discipline. It has seen its capabilities expanding as methods became more efficient and better approximated. Together with the advances in computational power, CEM has changed the nature of solving electromagnetic (EM) problems. There are

Code	Equation	Domain	Bandwidth	Suited Problems
MoM	Integral	Frequency	Narrow	PEC only. Not efficient for dielectric problems. Even worse with inhomogeneities.
FDTD	Differential	Time	Wide	Good for closed problems. Not as efficient as MoM at PEC only problems. Easily handles dielectrics.
FEM	Differential	Frequency	Narrow	Not as efficient as MoM in a PEC-only problem. Can easily handle dielectric problems.

Table 2.1: Brief comparison between the fundamental aspects of the discussed CEM codes.

many different methods available for computational modelling; MoM, FDTD and FEM (Finite Element Method) are principle examples. These codes can be grouped according to their different approaches to Maxwell's Equations [16, pp. 5-9]. MoM is based on discretisation of these equations in integral form. FDTD and FEM, on the other hand, are implemented using the differential form. These methods have also been applied using other formulations of Maxwell's equations. However, these other methods have not found widespread application. Table 2.1 gives a brief overview of the main differences between these three CEM codes. In this project, only MoM and FDTD were used and will, therefore, be the main topics of discussion.

2.2.1.1 Method of moments

MoM is the most widely used formulation that works by calculating equivalent currents from a radiating/scattering structure. Most implementations use a free-space Greens function. Boundary conditions are applied, and after that a linear set of equations is solved. This set then yields the solution to the surface currents of the meshed structure. The nature of this formulation makes it extremely efficient in PEC only problems. This is because the non-PEC regions do not have to be meshed, making it especially suited for antenna environments. The performance of MoM scales $O(f^6)$ ¹, where f is the frequency. This becomes worse when an inhomogeneous dielectric is added as expensive volumetric currents need to be calculated. For inhomogeneous dielectrics, the scaling approaches $O(f^9)$. To garner an understanding of how

¹Big O notation, which describes the behaviour of an algorithm according to its highest order term. This notation gives an indication of the asymptotic behaviour of the function

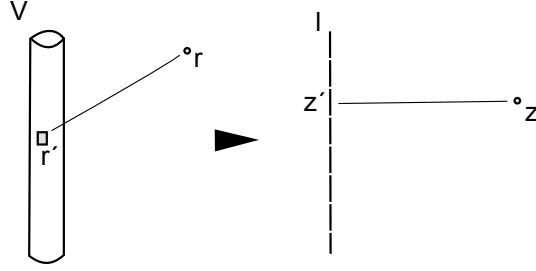


Figure 2.2: Diagram showing how the geometry of the wire example has been approximated into linear segments. The left side is the original problem where currents could exist in arbitrary orientations on the wire surface. On the right, currents are constrained onto the axial direction of the wire.

MoM functions at the most fundamental level, a simple thin-wire example is presented. This example can be found in literature covering MoM and was derived from Davidson [14, chap. 4].

Let us assume the problem of a straight wire in free-space. The potential anywhere on this wire can be calculated using Eq. 2.2.1.

$$V(\vec{r}) = \frac{1}{4\pi\epsilon_0} \int_V \frac{\rho(\vec{r}')}{R(\vec{r}, \vec{r}')} dV' \quad (2.2.1)$$

where

- r is the position being calculated
- r' is the position of the contributing source
- R is the distance between r and r'
- V is the volume of the wire
- ρ is the current density

To start, the problem itself can be reduced to simplify the example (see Fig. 2.2). The current on the wire is approximated only to exist in its axial direction. The current around the circumference of the wire is approximated to be evenly spread. For a thin highly-conductive wire, these are reasonable approximations. With the new formulation of the problem Eq. 2.2.1 can be simplified into Eq. 2.2.2

$$V(z, p = a) = \frac{1}{4\pi\epsilon_0} \int_0^l \frac{\rho_l(z')}{R(z, z')} dz' \quad (2.2.2)$$

where

- z is the position being calculated

- z' is the position of the contributing source
- R is the distance between z and z'
- l is the length of the wire
- a is the radius of the wire
- ρ is the current density

To this point, only the geometry of the problem has been simplified. The next step is to streamline the solution by first discretising the line into N segments. Each of these N segments has its own charge distribution. To describe this charge distribution, a basis function is needed. In this case, a square pulse is used which is unity over its associated segment and zero everywhere else. In Eq. 2.2.3 the current density can be seen described using these basis functions.

$$V(z) = \frac{1}{4\pi\epsilon_0} \int_0^l \frac{1}{R(z, z')} \left(\sum_{n=1}^N a_n h_n(z') \right) dz' \quad (2.2.3)$$

where

- a_n is the amplitude of the pulse describing the current density for this segment.
- h_n is the square-pulse basis function.

The amplitudes of the basis functions, a_n , are set as the current density in the middle of their respective segments. Because a_n is constant, it can be moved out of the integral. The form in Eq. 2.2.4 gives a better understanding.

$$4\pi\epsilon_0 V(z) = a_0 \int_0^\Delta \frac{h_0(z')}{R(z, z')} dz' + a_1 \int_\Delta^{2\Delta} \frac{h_1(z')}{R(z, z')} dz' + \dots + a_n \int_{(N-1)\Delta}^{N\Delta} \frac{h_n(z')}{R(z, z')} dz' \quad (2.2.4)$$

where

- Δ is the segment length.

The unknowns in this equation are the array of a_n variables. To solve these N unknowns boundary conditions are applied, and a linear set of equations is created by sampling in the middle of each segment. This set is then assembled into a matrix form and solved accordingly (see Eq. 2.2.5).

$$\begin{aligned} \mathbf{V} &= \mathbf{Z} \mathbf{I} \\ \mathbf{I} &= [\mathbf{Z}^{-1}] \mathbf{V} \end{aligned} \quad (2.2.5)$$

where

- $\mathbf{V}_m = 4\pi\epsilon_0 V(z_m)$
- $\mathbf{I}_n = a_n$
- $\mathbf{Z}_{mn} = \int_{(N-1)\Delta}^{N\Delta} \frac{1}{R(z,z')} dz'$

With this solution, the current density over the entire wire is known. With this information, it is possible to calculate the field at any point in space. This example demonstrates the basic functionality and use of a MoM code. There are multiple implementations of MoM; FEKO[®][17] is one of the well-known commercial distributions while NEC-2[18] is a popular public domain code. It can easily be seen why MoM is well suited to frequency domain problems involving PEC scatterers and radiators.

2.2.1.2 Finite difference time domain [14, chap. 2]

Another CEM model that has seen widespread adoption is FDTD. Different from MoM, it is formulated on the derivative versions of Maxwell equations. This lends itself to a very elegant and easy to implement form. This model solves problems by discretising an entire space into a grid. A very traditional solving strategy is using the Yee grid (see Fig. 2.4). Here the E-field and the H-field sample points are staggered. This grid is solved in a "marching on in time" fashion, meaning that no matrix is required. Field values at the next time step are a function only of the fields at this and previous steps. FDTD was adopted somewhat later than MoM. This is primarily a result of its handling of open boundaries, necessary computation power and the fact that problems were more suited to MoM in the earlier period. Today, FDTD has an extensive range of well-suited problems in industry, especially where dielectrics are involved. This was also accelerated by the invention of the Perfectly Matched Layer (PML), by Berenger in 1994, making the solution of open-problems very practical.

An FDTD algorithm has a similar starting point to that of a MoM solution. To begin the model is meshed. In this case, the whole volume including free-space is discretised. After that, the derivative equations are approximated using Taylor-series difference formulations with Eq. 2.2.6.

$$\frac{dU(x)}{dx} = \frac{U(x + \Delta x) - U(x - \Delta x)}{2\Delta x} - \frac{(\Delta x)^2}{12} \frac{d^4 U}{dx^4} + O(\Delta x)^4 \quad (2.2.6)$$

A simple transmission-line example is used to give some insight into the basic functioning of the code. Eq. 2.2.7 explains the propagation along a transmission line. This example was mainly derived from Davidson [14].

$$\frac{\partial I(z, t)}{\partial z} = -C \frac{\partial V(z, t)}{\partial t} \frac{\partial V(z, t)}{\partial z} = -L \frac{\partial I(z, t)}{\partial t} \quad (2.2.7)$$

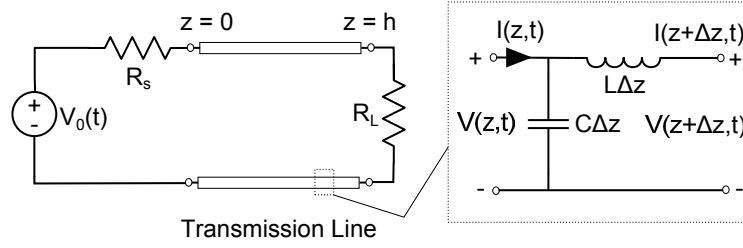


Figure 2.3: Schematic of the transmission line example. The transmission line itself has been discretised into lumped-element components.

The transmission line in Fig. 2.3 is meshed into N segments spanning from zero to h . This allows for a staggered grid to be created where V is calculated on the steps described by Eq. 2.2.8. I in this case is calculated on the steps described by Eq. 2.2.9.

$$\begin{aligned}\Delta z &= \frac{h}{N_z - 1} \\ \Delta t &= \frac{T}{M - 1}\end{aligned}\tag{2.2.8}$$

where

- Δz is the spatial step
- Δt is the temporal step
- h is the length of transmission line
- T is the length of simulation time
- $M > 2$

$$\begin{aligned}z_{\frac{1}{2}} &= \frac{1}{2}\Delta z \\ t_{\frac{1}{2}} &= \frac{1}{2}\Delta t\end{aligned}\tag{2.2.9}$$

An extra column and row are added outside of this grid to define the boundary and initial conditions. Initially, the entire transmission line will start discharged. Boundary conditions are defined by the excitation, V_o , and the terminating load. The final step before solving the grid is to define the update equations from Eq. 2.2.7. This is done using Eq. 2.2.6 with the resulting update equations in Eq. 2.2.10.

$$\begin{aligned}V_k^{n+1} &= V_k^n - \frac{\Delta t}{C\Delta z} \left(I_{k+\frac{1}{2}}^{n+\frac{1}{2}} - I_{k-\frac{1}{2}}^{n+\frac{1}{2}} \right) \\ I_{k+\frac{1}{2}}^{n+\frac{1}{2}} &= I_{k+\frac{1}{2}}^{n-\frac{1}{2}} - \frac{\Delta t}{L\Delta z} \left(V_{k+1}^n - V_k^n \right)\end{aligned}\tag{2.2.10}$$

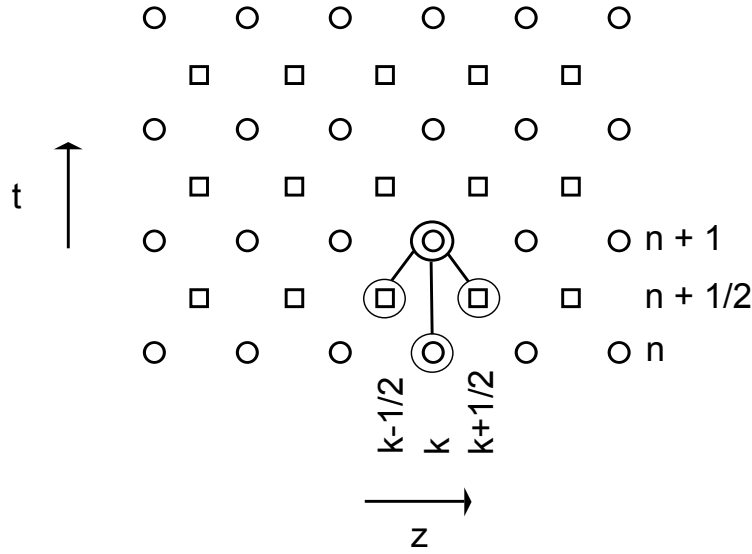


Figure 2.4: Diagram illustrating a 2D Yee grid for the transmission line example. Here the staggering of the voltage and current points can be seen. The variables involved in calculating a single voltage point are also demonstrated.

With these update equations, it is possible to solve the entire spatial/temporal grid given the boundary equations. These two update equations make it clear why FDTD is known for its simple implementation. However, much more effort is needed to solve complex problems. This illustrates the basic operation of an FDTD code. Traditional commercial implementations of these are XFDTD[®][19] and CST Microwave Studio (MWS)[®][20].

2.2.1.3 Geometrical optics [21, chap. 4]

Geometrical optics (GO) belongs to the class of asymptotic methods. These types of methods increase in accuracy with relative wavelength. Large high-frequency problems are well suited to ray-launched geometrical optics. The method lends itself to a very simple and intuitive formulation. GO has seen itself being used extensively on reflected antennas (compact ranges) and waveguides. It has also seen its use for propagation predictions over hilly terrain. However, the method itself is not without problems. Certain scenarios at caustic-ray points can cause infinite predictions that need to be solved by additional methods.

An understanding of GO is possible with a brief explanation. In the event where a plane wave is incident on the interface between two homogeneous media, a reflected, and refracted field can be calculated. Given a unity incident field at the interface, the two new fields can be described by Eq. 2.2.11.

$$\begin{aligned} E^r &= R e^{-jk_1 s^r} \\ E^t &= T e^{-jk_2 s^t} \end{aligned} \quad (2.2.11)$$

where

- E^r is the reflected ray
- E^t is the ray that passed through the interface
- k_1 and k_2 are the wave numbers associated with the media
- s^t and s^r represent the distance of the ray from the point of reflection and refraction
- T and R are the reflection coefficients and transmission coefficients

In some cases, the refracted field does not penetrate into the second medium. The latter is a function of the permittivity ratio between the two media. These are also calculated as a function of polarisation and dielectric parameters. In an extreme case where the interface is PEC, R becomes -1 for electric fields and 1 for magnetic fields. This is true for perpendicular polarisation only. The rays themselves are straight and represent the forward propagation of wave fronts. When these rays encounter an arbitrary surface, an assumption is made that the point behaves locally as if it was part of an infinite flat surface. The ray itself assumes at this stage the behaviour of a plane wave. In situations where these assumptions become reasonable and the media are weakly inhomogeneous GO becomes a practical solution.

2.2.2 Propagation and knife-edge diffraction [22, chap. 8][23][24]

2.2.2.1 Propagation models

Maxwell's equations state that a changing electric field produces a magnetic field, and a changing magnetic field produces an electric field. This behaviour is how electromagnetic energy can sustain propagation. An understanding of RF propagation can be regarded as mature, but there are complicated issues that are still to be resolved. Communication systems rely on the models produced by these studies for accurate planning. These models also become important in radio astronomy, where receiver systems are reaching ever-increasing levels of sensitivity. Because of the inherent complexity of environments, it is impossible to all but the simplest geometries to create deterministic models. Therefore, most models in wider use are of a statistical nature.

RF propagation is possible through line of sight (LOS) transmission, where there are no obstacles between the transmitter and receiver, as well as non-LOS transmission. The latter is made possible by reflection, refraction and diffraction. In the case of LOS, the fundamental limit for terrestrial propagation is the horizon. However, propagation is possible beyond this point through refraction as a result of the inhomogeneous atmosphere. This effect is accounted

for in many models using a $\frac{4}{3}$ correction for the radius of the earth. In effect, this refracted ray becomes straight [25]. In many practical situations, especially the environment investigated in this dissertation, non-LOS propagation is of interest. A brief explanation of the non-LOS propagation mechanisms follows:

- Reflection: The change of direction of the electromagnetic wave, at the interface between two media, back into its original media
- Refraction: The bending of electromagnetic waves due to inhomogeneities
- Diffraction: The bending of electromagnetic waves into the shadowed region of an obstacle

Non-LOS propagation is very dependent on the wavelength (see Eq. 2.2.12) relative to the obstacle size. Low frequencies tend to propagate over inter-continental distances due to ground-waves and sky-waves. However higher frequency signals can only propagate through LOS, making it easier to reuse these frequencies at different locations. A sky-wave is reflected off of the ionized plasma around the Earth. On the other hand, a ground-wave is guided by the temperature and moisture layers below 10 km. Table 2.2 summarises the different frequencies and their respective propagation mechanisms [26].

$$\lambda = \frac{c}{f} \quad (2.2.12)$$

where

- λ is the wavelength
- c is the speed of light
- f is the frequency of the wave

An example of the most simple propagation model is the Friis equation (Eq. 2.2.13) [30]. This equation is an entirely theoretical model accounting for only free-space propagation. In ordinary circumstances multi-path [31], ground-loss [32][33], atmospheric-loss [34], precipitation [35], foliage, tropospheric scattering [36][37] and gaseous absorption can also play a significant role.

$$L = G_T G_R \left(\frac{\lambda}{4\pi d} \right)^2 \quad (2.2.13)$$

where

- L is the total loss in the channel

Designation	Frequency	Mode of Propagation
VLF	3 kHz to 30 kHz	Ground-wave
LF & MF	30 kHz to 3 MHz	Ground-wave and Sky-wave
HF	3 MHz to 30 MHz	Sky-wave
VHF & UHF	30 MHz to 3 GHz	LOS and slight Sky-wave [27][28][29]
SHF	3 MHz to 30 GHz	LOS only
EHF	30 GHz to 300 GHz	LOS only (Gaseous attenuation becomes a problem)

Table 2.2: Summary of different terrestrial propagation mechanisms at various frequencies.

- G_T is the gain of the transmitting antenna
- G_R is the gain of the receiving antenna
- d is distance between the transmitter and receiver

The work in this dissertation focusses mostly on local-propagation effects. At these distances, gaseous absorption and atmospheric loss will not have any significant impact. Therefore, the focus is largely shifted to near-earth propagation. These near-earth models are mainly influenced by the terrain. A hilly terrain will cause shadowing, diffraction[38] and reflection effects. In the case where most of the ground is flat, ground loss and diffraction at the horizon will still be present. These models aim to give a measure of the median loss. Examples of some of the near-earth propagation models can be seen in Table 2.3.

2.2.2.2 Diffraction

The Huygens principle states that each point on a wavefront represents a source for a secondary wavelet. These combined wavelets produce a new wavefront in the direction of propagation (see Fig. 2.5) [45][46]. With this principle, it is easy to see how diffracted energy enters the shadowed region of an obstacle. Using diffraction models it is possible to deterministically calculate the field behind an obstacle. However, it is not easy to model diffraction for but the simplest of geometries. These include a wedge shape or a rounded wedge shape. In some cases, these shapes can be cascaded horizontally to form multiple obstructions. These calculations are only valid when the energy passing through the material is small (ideally PEC). Much of the work in this dissertation has to do with the shielding effectiveness of a berm structure. The shielding effectiveness of this structure is primarily affected by diffraction over and around it.

Name	Description
Egli Model[39]	Easy to implement. Does not address vegetation. Based on an empirical match to measurements.
Longley and Rice[40]	Uses a full terrain model. Also takes into account climate, sub-soil and ground curvature. Applied in the form of software (SPLAT).
ITU Model[41]	Based on diffraction theory and the height of the obstruction.
Built-up Area models	Young[42], Hata[43] and Lee[44] are all empirical models that are used in varying built-up areas.

Table 2.3: Examples of some near-earth propagation models.

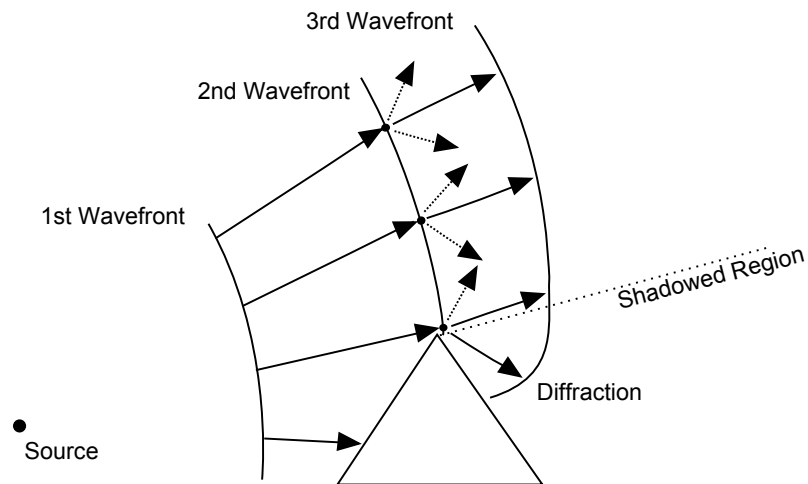


Figure 2.5: Illustration of Huygens principle. A source emits a wavefront into a wedge obstacle. The third wavefront demonstrates how energy is diffracted into the shadowed region by using Huygens principle.

2.3 Existing Multi-copter-based metrology vehicles

To investigate existing devices, we need to define what constitutes as a Multi-copter RF metrology vehicle. For the purposes of this dissertation, such a vehicle will be defined as follows:

- Has the capacity of autonomous flight given a set of pre-programmed way-points.
- Contains a detector or detectors capable of measuring or transmitting a subset of the electromagnetic spectrum.
- Uses some form of an antenna or antennas to transduce the electromagnetic energy into signals compatible with its detector or the other way around.

Multi-copters have been growing in popularity as their accessibility and reliability have improved. It is possible to buy off-the-shelf flight controllers that have the capability of autonomous flight. These technologies have made Multi-copter vehicles an attractive platform for various applications, such as electromagnetic metrology. Their autonomy and inherent access to arbitrary heights open up new possibilities. 3D far-field patterns can be measured at on-site locations such as broadcasting towers. Propagation measurements are possible at previously unattainable altitudes, and near-field scans of antenna arrays have also been done. In Kibet-Langat [6] and Groch [5] SKA power lines end-fire patterns were measured using a spark generator. A South African company (LS of SA), part of LS Telcom (see Fig. 2.6), has their own fleet of Multi-copters which are used for surveying broadcasting towers and cellular masts [47]. These surveys were previously only possible using human-piloted vehicles. This made measurements much more expensive and in most cases impractical. At the University of Nijmegen and IEIT, gain calibrations have been investigated on LOFAR and SKA low elements using a similar type of Multi-copter vehicle [48][49][50][51].

Multi-copters offer an attractive set of advantages over the conventional ground, mast or human piloted aircraft measurements. However, as a small and relatively light aircraft, these advantages come at a price. The size and weight of onboard equipment plays a significant role in the cost and performance of the vehicle. Also, in the electromagnetic domain, the geometry of the Multi-copter has a significant impact on its metrology antennas.

Multi-copter metrology vehicles can be broken up into two main branches, sources and receivers. The University of Nijmegen and IEIT are making extensive use of a source type configuration for characterising antenna arrays. In this case, the vehicle is used to replicate a far-field source by transmitting a fixed-carrier frequency. This work is also being applied to the SKALA array



Figure 2.6: Photo of LS of SA using their Multi-copter platform for a measurement. Receiver and computer equipment are mounted on the bottom of the vehicle with the antenna secured to a fixture between two of the motor arms.



Figure 2.7: Photo of the IEIIT Group Multi-copter during an array scanning measurement. Dipole antenna and tone generator can be seen mounted on its underside.



Figure 2.8: Photo of a Multi-copter being used at the University of Nijmegen. The antenna, in this case, has been placed away from the vehicle. Photo was taken from Krause [48].

configurations. The source type Multi-copter from the IEIIT group can be seen in Fig. 2.7. A signal generator feeding a dipole is mounted at the bottom of the aircraft. The more common type of configuration using the vehicle as a receiver is used by LS Telcom and other parties to survey the performance of broadcasting and cellular installations. Fig. 2.6 shows LS Telcom's vehicle during flight. Because this vehicle is configured as a receiver, a spectrum analyser and a single-board computer need to be mounted onboard. A stub antenna, in this case, is mounted on a fixture between two of the motor arms. In both these cases, the complex structure of the vehicle influences the properties of the antennas. Therefore, the antennas need to be characterised onboard the vehicle with care taken not to disturb the metallic environment afterwards. The vehicle from Nijmegen, seen in Fig. 2.8, attempts to move the antenna away from the vehicle to minimise its influence.

The source and receiver branches can be further broken into frequency domain or time domain arrangements. All of the afore-mentioned vehicles use a frequency domain approach leading to relatively narrowband measurements. Currently, vehicles using a time-domain scheme do not exist. Such a configuration would greatly increase measured bandwidth and would thus reduce measurement time.

2.4 Dielectric measurements

2.4.1 Dielectrics and their constitutive parameters

The field of Dielectric properties measurement is relatively well established and has been around for some time. Many methods are available for both the analysis and extraction of dielectric parameters. The information in this subsection was derived mainly from Balanis [52, chap. 1 and 2], to give a brief introduction into dielectric metrology. Maxwell's equations are studied, followed by an overview of some typical measurement systems, and to end with, a quick look into the extraction methods used to find dielectric parameters from their measurements.

It is useful to know what a dielectric is and how it is affected by electromagnetic energy. A dielectric can be defined as having the property of transmitting electrical force without conduction. This is illustrated by the last term in Eq. 2.4.1, which is also the famous displacement current term that Maxwell added to Amperes law, with J electric current density, H and E representing magnetic and electric fields respectively.

$$\nabla \times H = J + \frac{\partial \varepsilon E}{\partial t} \quad (2.4.1)$$

This term derives from a form distinguishing between the effect of free-space and polarisable mediums as seen in Eq. 2.4.2. Assuming harmonic time variation the latter equation is simplified to Eq. 2.4.3. The part concerning the material properties can be lumped into a single constitutive parameter ε^* , seen in Eq. 2.4.4.

$$\nabla \times H = \sigma E + \frac{\partial P}{\partial t} + \frac{\partial \varepsilon_0 E}{\partial t} \quad (2.4.2)$$

$$\begin{aligned} \nabla \times H &= \sigma E + j\omega(P + \varepsilon_0 E) \\ \nabla \times H &= j\omega\left(\varepsilon_0 + \frac{P}{E} - j\frac{\sigma}{\omega}\right)E \end{aligned} \quad (2.4.3)$$

$$\nabla \times H = j\omega\varepsilon^* E \quad (2.4.4)$$

There are two terms, P and σ , which determine the constitutive parameter ε^* . Conductivity is represented by σ in Siemens per meter. P , on the other hand, is slightly more involved and will be briefly explained.

The moment a material is subjected to an electric field, the material will react by becoming polarised. When the electric field is removed, the material will again depolarise over time. A simple dipole moment is illustrated in Fig. 2.9, which can be described by Eq. 2.4.5, where Q represents charge, l_i

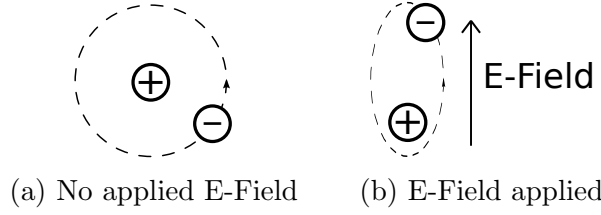


Figure 2.9: An illustration of a material subject to an applied E-field. After [52].

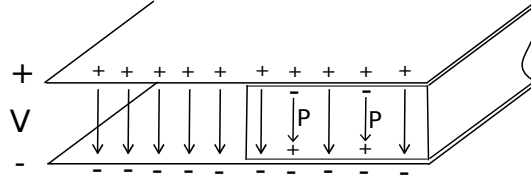


Figure 2.10: Parallel plate illustration. After [52].

the distance between the two charges and finally dp_i the elemental polarisation moment.

$$dp_i = Ql_i \quad (2.4.5)$$

When all of the dipoles throughout a medium are added together (Eq. 2.4.6) and divided by its material volume, the polarisation vector, P , is found. P represents the stored energy of the material in the previous Eq. 2.4.3.

$$P = \lim_{\Delta v \rightarrow 0} \left[\frac{1}{\Delta v} \sum_{i=1}^N dp_i \right] \quad (2.4.6)$$

The effect of the polarisation vector can be illustrated visually using parallel plates as seen in Fig. 2.10. In this example, an electric field is applied between two plates in a space where only half of the space is filled with a dielectric material. The electric-flux density in the free-space region is described by Eq. 2.4.7, where E_a is the applied electric field.

$$D_0 = \varepsilon_0 E_a \quad (2.4.7)$$

In the material region of the parallel plates, the electric-flux density changes with the addition of the polarisation vector. The polarisation vector tends to cancel out the charge on the two plates. However, because the source needs to maintain the electric field between the plates, the cancelled charges are replaced by new charges. Therefore the electric field between the plates, E_a , stays unchanged while the flux density, D , increases because of the material polarisation. This is also described in Eq. 2.4.8.

$$D = \varepsilon_0 E_a + P \quad (2.4.8)$$

The polarisation vector can also be written in another form as seen in Eq. 2.4.9. This introduces the dimensionless term electric susceptibility, χ_e . This term when added back into Eq. 2.4.8 leads to a familiar equation seen in Eq. 2.4.10.

$$P = \varepsilon_0 \chi_e E_a \quad (2.4.9)$$

$$D = \varepsilon_0(1 + \chi_e)E_a = \varepsilon_0 \varepsilon_r E_a \quad (2.4.10)$$

Most materials, especially the material being measured, will have not only a real permittivity, but also a lossy component represented as a complex term, ε_r^* . Complex permittivity, ε_r^* will be seen throughout this dissertation in a simplified form (see Eq. 2.4.11). ε_r^* can be broken into its complex constituents where ε' represents the real part that relates to the stored energy in the medium and ε'' represents the imaginary part that relates to the dissipation and loss. A similar parameter, μ_r^* , describes the effect of the magnetic field on the material. The dielectrics used in the measurements will not have any noticeable magnetic properties and μ_r^* will, therefore, be taken as unity. These constitutive parameters make it possible to describe the response of a material to external electric and magnetic field. However, to use these parameters they have to be measured. A brief overview of the systems employed in these measurements will be given in the next section.

$$\varepsilon_r^* = \varepsilon' - j\varepsilon'' \quad (2.4.11)$$

2.4.2 Dielectric measurement techniques

Numerous methods exist to measure dielectrics, each with their benefits and drawbacks [53]. Three different techniques were used during this dissertation: free-space transmission system, guided-wave transmission system (waveguide) and a probe-reflection system. These systems were used for the frequency range they cover as well as their immediate availability. Each system will be briefly discussed in its subsection. Also, a simple comparison of the three measurement systems can be seen in Table 2.4.

2.4.2.1 Free-space transmission measurement system

The free-space transmission system consists of two antennas directed at each other focusing on a sample holder area seen in Fig. 2.11. After calibration, the sample is placed in the middle of the two antennas. These systems are high-frequency, band specific and prefer thin samples in order to preserve sensitivity and negate the effect of the local antenna patterns. They are ideal for measurements requiring sample temperature variation and are less sensitive to material preparation.



Figure 2.11: Free-space measurement system.

2.4.2.2 Waveguide transmission measurement system

The waveguide system consists of two coaxial-to-waveguide feeds which attach to opposite ends of a material holder section. This section is just another piece of waveguide that is filled with the intended material, see Fig. 2.12. These systems are very sensitive to material preparation where the material has to fill the sample holder entirely with a flush surface on either side. Another drawback is the time it takes to do the calibration. Material exchanges tend to take a significant amount of time due to the mechanical nature of attaching and removing waveguide sections. The waveguide measurement systems are also subject to band limitations depending on their dimensions. For simplicity, measurements were only made using the TE_{10} (Transverse Electric) mode of the particular waveguide.

2.4.2.3 Coaxial-probe system

The coaxial-probe is a non-invasive system which allows for easier measurement. However, some information is lost because the transmission S-Parameter, S_{21} , is omitted. A coaxial-probe developed in house [54] can be seen in Fig. 2.13. Because this is a coaxial system, the measurements are usually broadband. These systems require a known material to be measured for extraction to be possible.

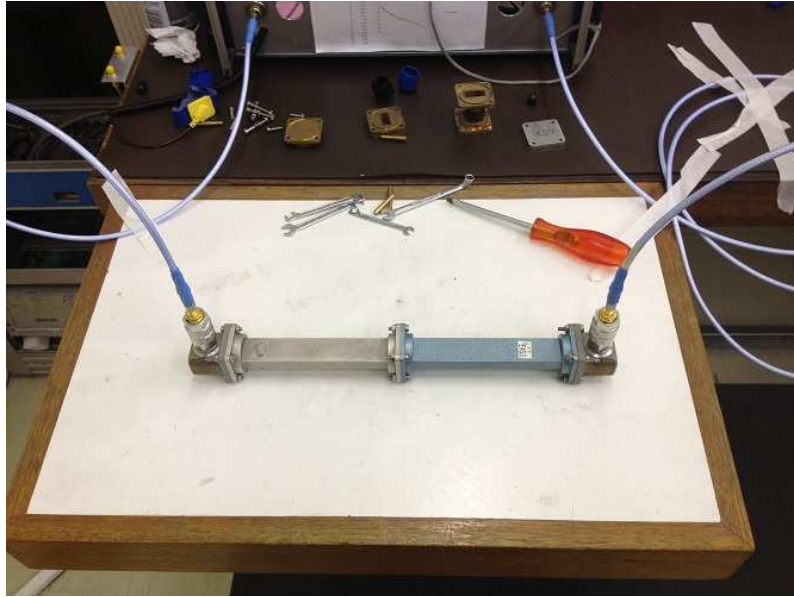


Figure 2.12: X-band waveguide measurement system during a through calibration. Material holders and calibration sections can be seen in the background.

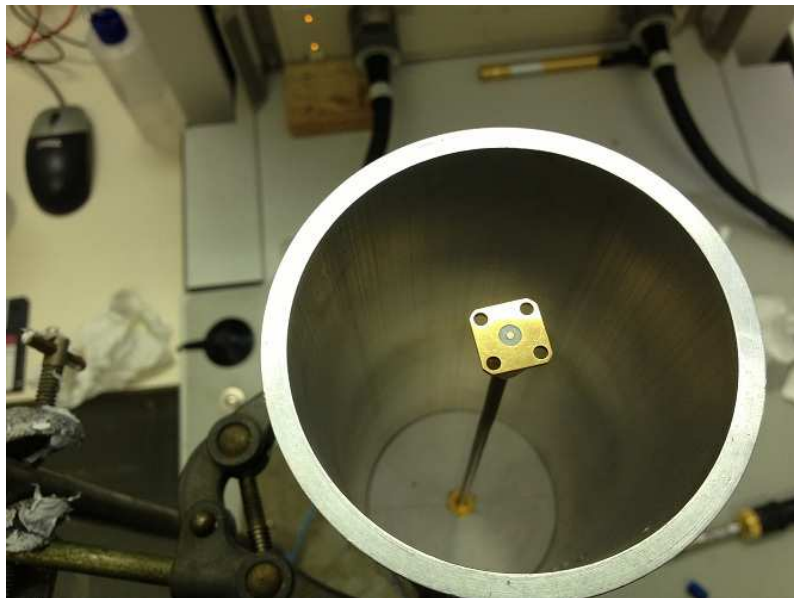


Figure 2.13: Coaxial probe system.

	Free-space	Waveguide	Coaxial
Frequency range	Band limited	Band limited	Broad band
Sample Preparation	Easy	Difficult	Easy
Measured Parameter	ε, μ	ε, μ	ε
Impact on test material	Non-destructive	Destructive	Non-destructive

Table 2.4: Comparison table between three different dielectric measurement methods used in this dissertation [53].

2.4.3 Parameter extraction techniques

After measuring scattering parameters with the respective measurement systems, a suitable extraction technique is required to convert the scattering parameters into their permittivity and permeability data. A well-known method, of which a variant is used in this dissertation, is the Nicolson-Ross-Weir (NRW) method. The NRW method is a fast-analytical approach which inverts the scattering parameters into their constitutive parameters ε^* and μ^* [55][56]. For the latter inversion, both the S_{11} and S_{21} parameters are needed. Therefore, this method is more applicable to the waveguide and free-space measurements. For the coaxial system a different method is used, discussed in Rimbi [57].

The NRW method has a limitation. As soon as the wavelength in the sample is shorter than the sample itself, the extracted data can become invalid. This problem will be encountered in this text and will be dealt with using different tool-sets. There exists an iterative method, Baker-Jarvis, which solves multiple problems seen in the NRW method including the wavelength phase ambiguity and the unstable $\frac{1}{S_{11}}$ term at half-wavelength intervals [58]. However, for the purposes of this work, an improved version of the NRW method was sufficient.

2.5 Conclusion

The focus of this chapter was to give a background to key topics discussed in the dissertation. The literature study started with an overview of the South African SKA effort and how RFI and propagation studies form an essential part of the project. A brief overview was given into different areas of computational modelling, including MoM, the FDTD method as well as GO. After that, some background was given in propagation, diffraction and knife-edge models. Finally, some attention was given to current Multi-copter developments followed by some theoretical and practical background to dielectric measurements. The next chapter will shift its focus to the broadband characterisation of a dielectric block that will be used extensively throughout the scale modelling later in this dissertation.

Chapter 3

Dielectric Measurement

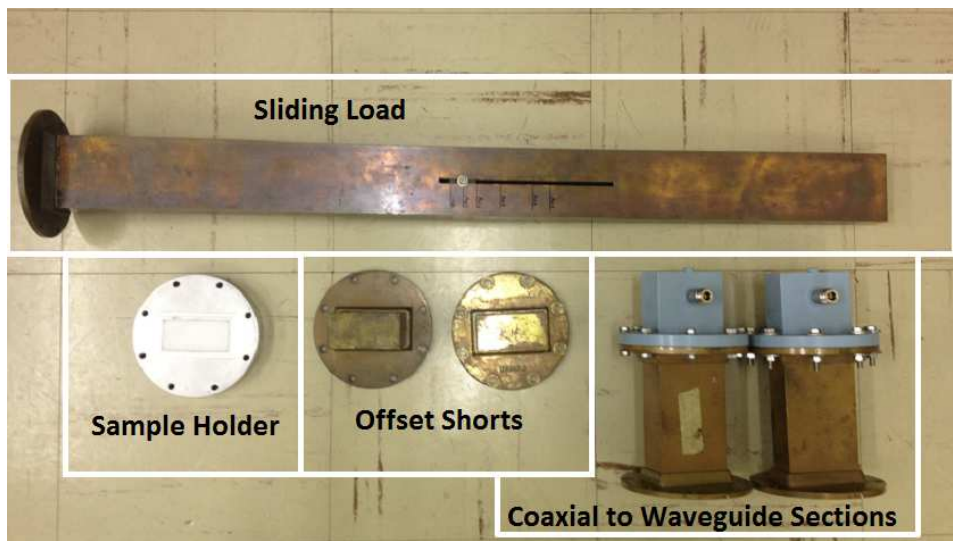
3.1 Introduction

This chapter will focus on the characterisation of a carbon-impregnated foam block. The permittivity will be of importance when using the material in laboratory-scale measurements of Chapter 4. Three different transmission and reflection methods were utilized in the characterisation; waveguide, free-space and coaxial. The latter was not used in the final model due to the unreliability of the measurements. This chapter will discuss each measurement of which the results will be gathered and fitted to an analytical curve. This curve will be used as the dielectric model in CEM packages. The characterisation will be realised over the frequency range of 2 GHz to 17 GHz. The lower limit has been determined by the scale modelling while the upper limit was determined by the highest frequency of the free-space measurement system. Therefore, the fitted model will be used to infer the permittivity up to 20 GHz.

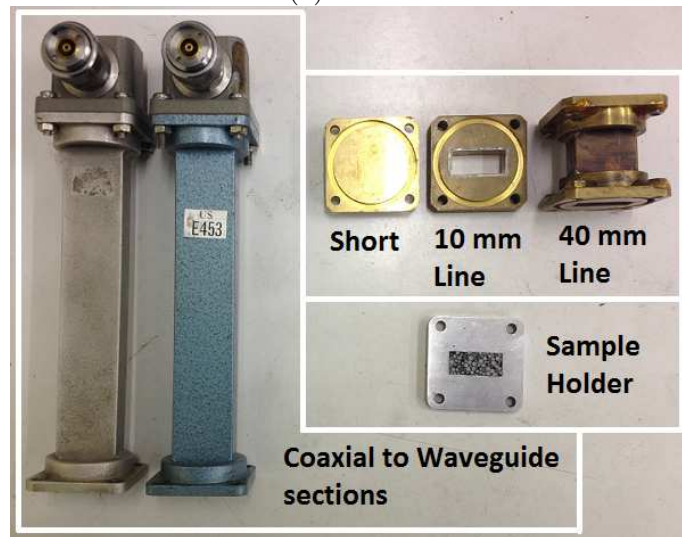
3.2 Waveguide measurement

The first apparatus is a waveguide system. These measurements are bounded in bandwidth by the physical dimensions of the waveguide. Because of this, two waveguide sets were used with their frequency ranges situated in the S and X-band sections. The calibration, measurement and extraction for these two systems are identical and will be discussed shortly.

The measurement sets can be seen in Fig. 3.1 which includes the calibration standards and sample holders. A brief overview of the physical aspects of these systems has been listed in Table 3.1. To ensure reliable measurements, it is of importance that the material is properly fitted to the sample holder. This meant taking care that the faces are flush against the interfaces and eliminating any cavities. Fig. 3.2 shows a sample holder along with three S-Band samples; carbon impregnated foam, Teflon and Perspex. The well characterised Teflon and Perspex samples will be used to check the reliability of the measurement.



(a) S-band



(b) X-band

Figure 3.1: Photographs of the S-Band and X-Band waveguide measurement sets including calibration standards and sample holders.

	S-Band	X-Band
Guide width (a)	72.14 mm	22.86 mm
Guide height (b)	34.04 mm	10.16 mm
Sample holder length (d)	25 mm	10 mm
Frequency Band TE_{10}	2.1 - 4.1 GHz	6.6 - 13.1 GHz
Calibration method	2 shorts and a sliding load	Short, line and thru

Table 3.1: Overview and comparison of S and X-band waveguide systems.

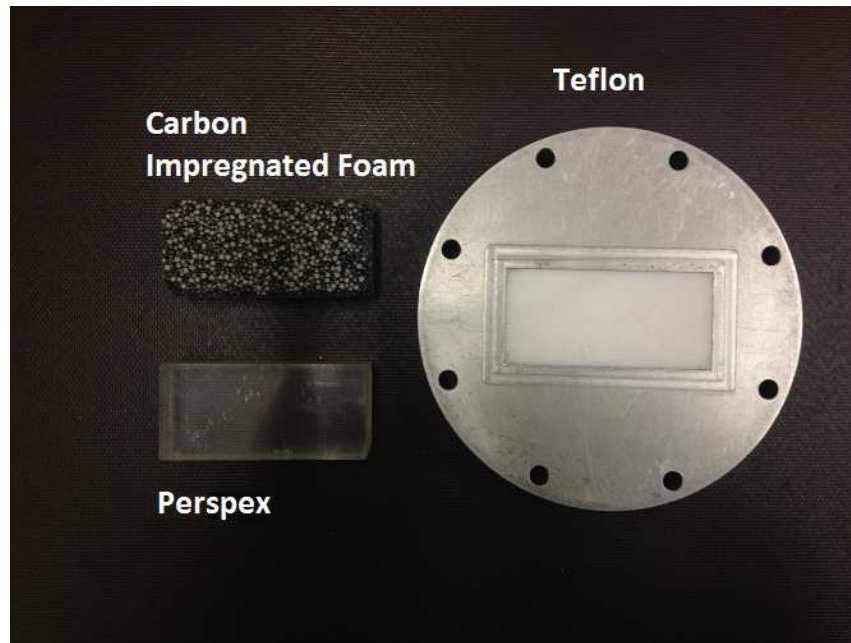


Figure 3.2: S-Band samples with Teflon already fitted into the sample holder. Teflon and Perspex will be used for reference measurements.

A demonstration of a typical waveguide measurement can be seen in Fig. 2.12. From this, it can be seen how a measurement set consists of two coaxial-to-waveguide transition sections. These transition sections are usually at least two wavelengths long, measured at the lowest measurement frequency. This is to negate the effects of the coaxial-to-waveguide evanescent modes in the sample cavity. Finally, the sample holder is located in the middle between the defined calibration planes.

The calibration planes are defined during the waveguide measurement's calibration sequence. This plane is in most cases the plane at which the calibration standards are connected. These are the reference positions where phase and magnitude are defined as 0 degrees and a magnitude of 0 dB respectively. The calibration standards for these measurements can be found in Appendix A. Measurements were made in a temperature stable room in the shortest amount of time possible to preserve calibration. After a successful calibration, the S-parameters of the foam could be measured.

3.2.1 Measurement

The calibration of the S-Band measurements was first validated using the known samples, see Appendix A.2. In the case of the X-band system, such reference samples were not available. However, with the successful process used in the S-Band system it was inferred from several repeatable measurements that the X-Band calibration was successful.

Fig. 3.3 and 3.4 shows the S-parameters measured respectively on the S-Band and X-Band samples. In each plot magnitude, as well as phase, has been represented with a dual y-axis. The fundamental, as well as higher-order mode cut-in responses have been shifted out of the plot range with a frequency buffer of 10%. In these measurements time-gating¹ was not used to suppress secondary reflections. However, it should be noted that the use of time-gating helps reduce the unwanted ripple due to secondary reflections. It is good practice to confirm that the phase does not have a positive trend, which is an indication that the sample is misaligned into the reference plane. Also, it is worth making sure, that for the given sample holder length, the permittivity of the sample does not cause the wavelength in the sample to be shorter than the actual sample holder. If this is the case, the current NRW extraction method will not work, the reason for this will be made clear in the next section. In this case, because the material was used for a previous study [59], we are sure that the material has a low enough permittivity for the length of our sample holders. This has been determined from Eq. 3.2.1 where f_c is the cut-off frequency for the TE_{10} mode, λ the free-space wavelength and f the lowest frequency used during extraction (Chapter 7 in [52]). For the NRW algorithm to work reliably the wavelength criterion in Eq. 3.2.2, where d is the sample holder length, must be satisfied. At this point, the measured S_{11} phase will become unstable and cause inaccurate extraction. λ_g is the wavelength inside the waveguide.

$$\lambda_g = \frac{\lambda}{\sqrt{1 - \left(\frac{f_c}{f}\right)^2}} \quad (3.2.1)$$

$$\lambda_g > 4d \quad (3.2.2)$$

Finally with these successful measurements it is possible to extract the material parameters.

3.2.2 Material parameter extraction

With the S-Parameters, a NRW method is used to extract the permittivity and permeability of the material. The NRW method was discussed briefly in Section 2.4.3. Although extremely fast, this method does not take into account its prior measurement points, therefore it is unable to detect if the phase wrapped² in the measured frequency band. This causes the algorithm to always use the fundamental root illustrated as $\phi(n = 0)$ in Eq. 3.3.4. Therefore, an improved NRW algorithm was used, derived from Luukkonen, Maslovski and Tretyakov [60]. This method unwraps the phase before extraction and

¹Selecting only portions of a signal between specific time intervals before transforming into the frequency domain.

²The measured phase is always constrained to its principle interval $[-180, 180]$.

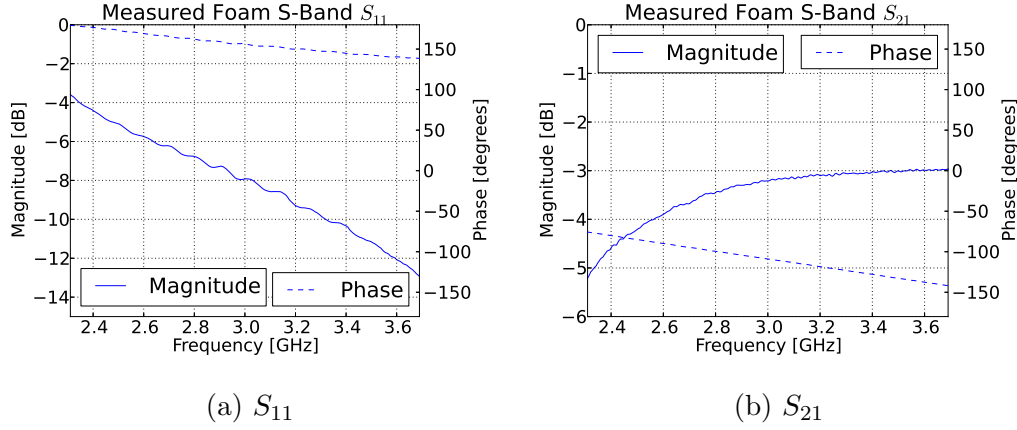


Figure 3.3: S-Band waveguide time-gated S-parameters.

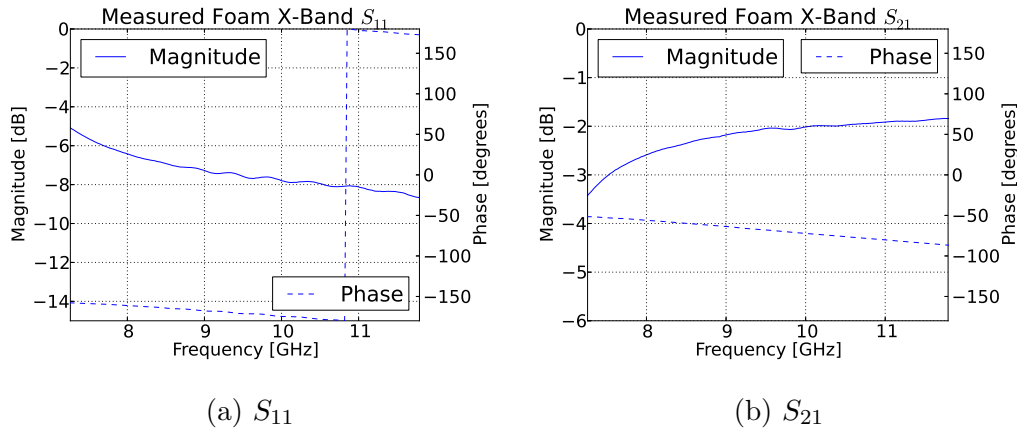


Figure 3.4: X-Band waveguide time-gated S-parameters.

uses this new phase vector to do its calculations. This algorithm however still demands that the first measured point still be part of its fundamental root which will be the case in all of the waveguide measurements. Appendix A.3 has an illustration of using the incorrect root in extraction and showcases the advantage of the improved NRW method.

With the improved NRW extraction method, the S-parameters in Fig. 3.3 and 3.4 were extracted as seen in Fig. 3.5. Because of the nature of the materials used, the measured permeability is all close to unity. For simplicity and the sake of space, the permeability of all the materials in measurements will be taken as one throughout their frequency range.

In both extractions, the measured permittivity of the foam has a relatively constant trend, where $\epsilon = 2 - 0.48j$ would be a good approximation through the frequency range. Teflon shows the same stable trend while Perspex starts to deviate at frequencies above 3.5 GHz, which will be made clear in a moment. The average value, for Teflon and Perspex, from the stable part of each

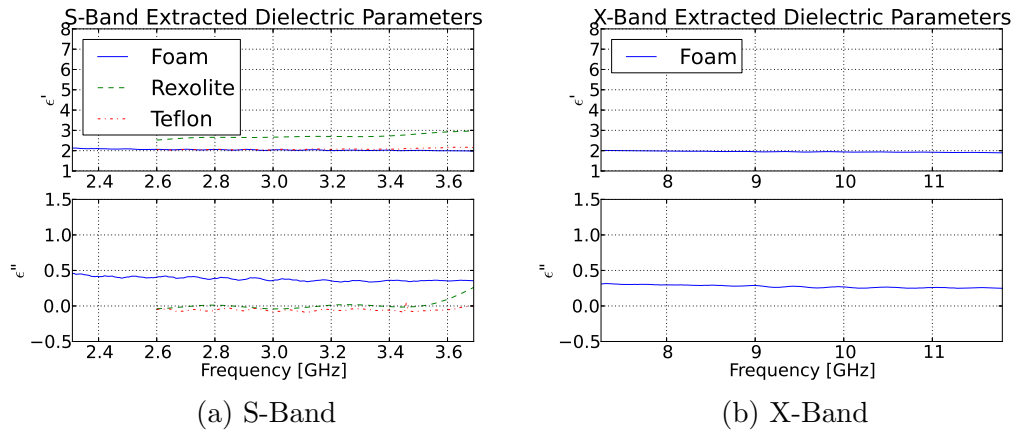


Figure 3.5: S-Band and X-Band permittivity extracted with the improved NRW algorithm.

	Measured	Known	Frequency Band of Ref. Material
Perspex	2.65	2.6 [61]	10 kHz to 3 GHz
Teflon	2.05	1.96 [62]	100 Hz to 25 GHz

Table 3.2: Comparison between known and measured values for Perspex and Teflon. The reference values are both average values over their given frequency bands.

measurement has been tabulated in Table. 3.2. Here the previously measured values are compared to the latest measurements to check the reliability of the extraction.

The deviation for Perspex at 3.5 GHz has to do with the fact that the measured Perspex, deduced from the constant trend before the diversion, has a permittivity of 2.65. This is too high for this frequency and sample-holder length combination. Calculating the new guide wavelength using Eq. 3.2.1, it can be found that $\lambda_g = 9.2 \text{ cm}$. With a sample holder length of 2.5 cm it is clear that the criterion in Eq. 3.2.2 has not been met. This means that the measured S_{11} term became small as it was moving through its null point, causing instabilities in the NRW extraction seen in the Perspex measurement.

It can be seen in Table 3.2 that the measurement was within 5% of the reference values giving confidence in the extraction. The extracted values for the foam will be used again at the end of this chapter after the rest of the frequency band has been extracted. The next section will focus on a higher frequency band using the free-space-measurement system. The code used in this section to extract the permittivity from the measurements can be found in Appendix A.3.

3.3 Free-space measurement

In this section, the dielectric is measured using a free-space system. The available system [63] operates at a frequency of 11 to 17 GHz. Together with the waveguide measurements, this covers the greater part of the 2 GHz to 20 GHz band needed for the laboratory-scale measurements. The free-space system seen in Fig. 2.11 consists of two horn antennas focussed at a centre point. This is also where the material under test (MUT) is situated.

The measurement and extraction process is similar to that of the waveguide system. However, calibration in this case will be done during post-processing. Due to the length of the measurement samples, another step will be added to the extraction. This additional step of finding the correct measured phase, or root, is due to the phase ambiguity. The latter is a consequence of the sample length being longer than the wavelength in the sample itself. The width of the samples could not be changed as the samples themselves could not be destroyed or altered. As a result, the measured samples are somewhat large for this apparatus. This causes the reference planes to be located slightly closer to the horn antennas and out of the focused region. The extraction assumes a normally-incident transverse electromagnetic (TEM) wave propagating through the MUT. An in-depth analysis in [63, chap. 5] studies the fields present in the focal region of the antennas. This study deems the fields centred 20 mm around the focal plane has satisfactory properties to be assumed as an incident plane wave. The material being measured here has a width of 115 mm. Examining the field in the focal region of the system, it could be seen that variations are limited to a maximum of 2 dB. Therefore, it was deemed that the measurements will still be sufficient.

3.3.1 Measurement and calibration

To start, a 2-port calibration was completed up to the waveguide interfaces³. This allowed verification, in the time domain, of all the system's known reflections and transmissions. The latter included finding the position of the focal region in time, which enables the use of time-gating. Time-gating in this system will help reduce reflections from the metallic-antenna structures. The actual offline calibration consists of the standard calibration measurements and storing them to be used in post-processing with the real MUT measurements. The calibration used in this case was a load, short and offset short calibration. The positioning of the offset short was achieved using the built-in precision positioner. The short itself was simply a metallic plate as seen in Fig. 3.6. The load that was used can be seen in the same figure behind the metallic plate, in this case to negate extra reflections.

³The horns on the free-space system are fed through Ku-band waveguide sections

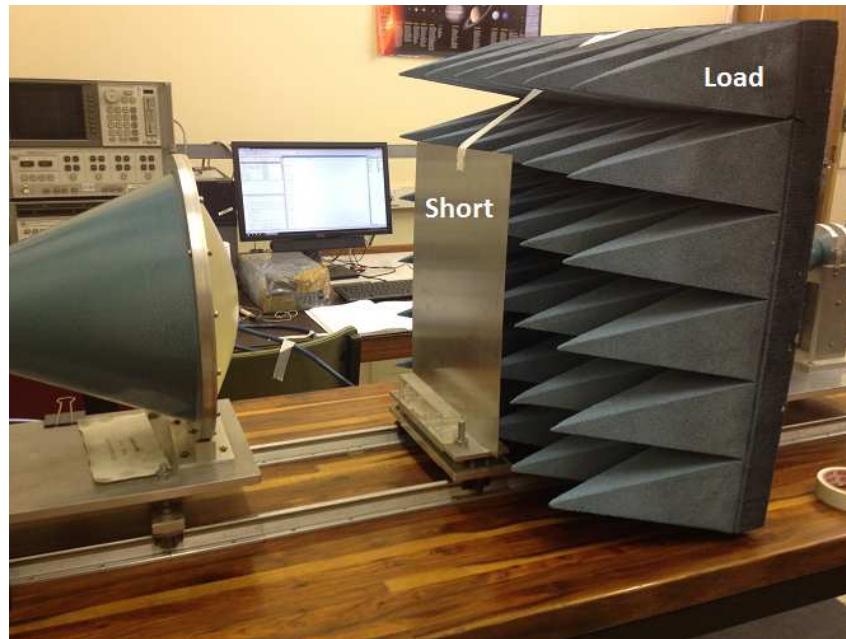


Figure 3.6: Offline calibration using a load, short and offset short. At this point the short is measured at $z = 0$ with an absorbing foam block, also the load, behind it.

	Measured Trace	Standard Definition
Load	M_1	$S_1 = 0$
Short	M_2	$S_2 = -1e^0$
Offset Short	M_3	$S_3 = -1e^{-j2\beta_a 5.7e-3}$

Table 3.3: Calibration standard definitions for the free-space measurement system. The measured traces for each standard will be named according to M_1, M_2 and M_3 .

For the offline calibration, three standards were measured as mentioned before. Here variables are assigned for clarity in the equations, see Table 3.3. β_a in this case is the adjusted wave number because of the lens effect where $\beta_a = \frac{\beta}{1.045}$. This adjustment term, as well as the rest of the equations used in the offline calibration, comes from [63].

The idea of the offline calibration, as well as any other calibration, is to remove systematic errors, errors that are repeatable, from the measurements. Two other errors that are not removed by calibration are random errors and drift errors. The former can be reduced by averaging measurements while the latter, which is due to pressure and temperature drift, can be reduced by doing measurements in the shortest timescale possible. To further explain the offline calibration process, a model for the systematic errors can be seen in Fig. 3.7 [64]. This model only accounts for a one-path calibration.

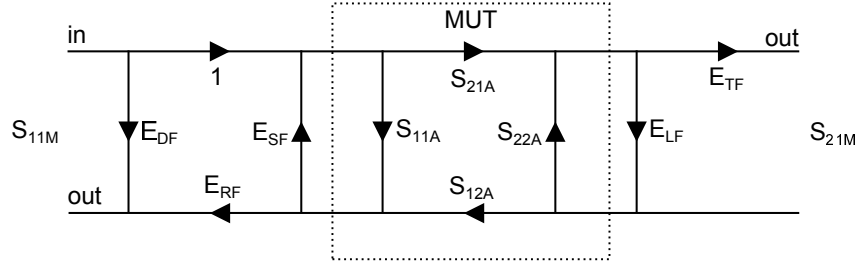


Figure 3.7: Model of S-parameter systematic errors.

Error	Expression
Directivity Error	$E_{DF} = M_1$
Reflection Tracking	$E_{RF} = \frac{(S_2 - S_3)(M_1 - M_3)(M_1 - M_2)}{S_2 S_3 (M_3 - M_2)}$
Source Match	$E_{SF} = \frac{S_3(M_2 - M_1) - S_2(M_3 - M_1)}{S_2 S_3 (M_3 - M_2)}$
Transmission Frequency Response	$E_{TF} = S_{21M}$
Load Match	$E_{LF} = 0$

Table 3.4: Expressions for the error coefficient in the model seen in Fig. 3.7.

Using this error model the actual S-parameters can be expressed in terms of the measured S-parameters and the errors present in the system seen in Eq. 3.3.1 and Eq. 3.3.2. E_{LF} was assumed to be zero to make the measurements more practical. This assumption might cause some ripple in the calibration, however, when time-gating is applied, the assumption becomes valid.

$$S_{11A} = \frac{S_{11M} - E_{DF}}{E_{SF}(S_{11M} - E_{DF}) + E_{RF}} \quad (3.3.1)$$

$$S_{21A} = \frac{S_{21M}(1 - E_{SF}S_{11A})}{E_{TF}} \quad (3.3.2)$$

Using these equations, expressions can be developed for each of the error terms listed in Table 3.4. The measured S-parameters, along with the error terms, calculated according to Table. 3.4, can be seen in Appendix B.2. Using each of these error terms along with the measured S-parameters, the offline calibrated S-parameters can be found with the expressions in Eq. 3.3.1 and Eq. 3.3.2. The calibrated S-parameters for the foam material can be seen in Fig. 3.9. The sample used in the analysis has a thickness of 115 mm and a width and height of 420 mm and 210 mm respectively. The thickness of the sample is advantageous in that it increases the sensitivity of the measurement, however, in this case the focus was deemed useful 20 mm centred on the focal point. This might cause some inaccuracies in the extraction. To minimize these errors, the material was placed centred around the calibration plane. The calibration plane was then shifted in post-processing to the material face.

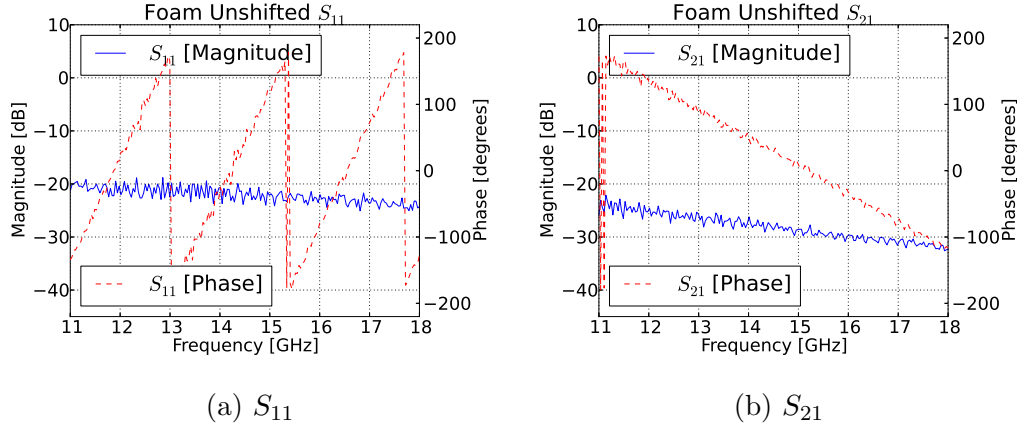


Figure 3.8: Offline calibrated measurement of the foam on the free-space system.

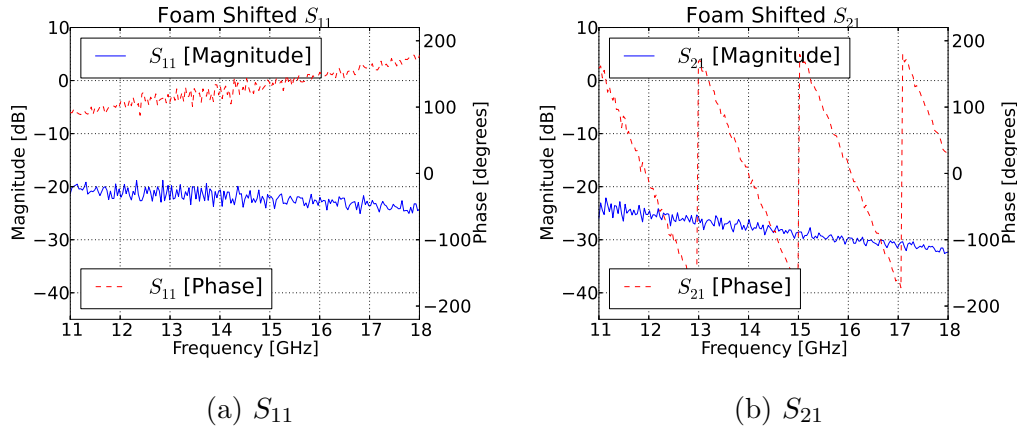


Figure 3.9: Calibrated S-parameter measurements of the foam with reference planes shifted.

3.3.2 Material-properties extraction

From here the extraction process would have been identical if the sample was thinner than a half wavelength. Because this is not the case, an extra step is needed to find the correct starting phase in the extraction. This is also known as choosing the proper root. By simply choosing the fundamental root, $n = 0$, the phase will be underestimated. To illustrate, Eq. 3.3.3 shows the refraction calculation step in the extraction. This step contains the $\ln(e^{\gamma d})$ term, which is responsible for the root problem. This term is expanded in the stepwise-phase method as seen in Eq. 3.3.4, where n signifies the root. It is possible to solve for the correct root by repeating the measurement for two different lengths of the same material [65]. The latter is accomplished by extrapolating the phase backwards to the point where they match with minimum error. This is done using Eq. 3.3.5, where p_1 and p_2 are the roots of the measured

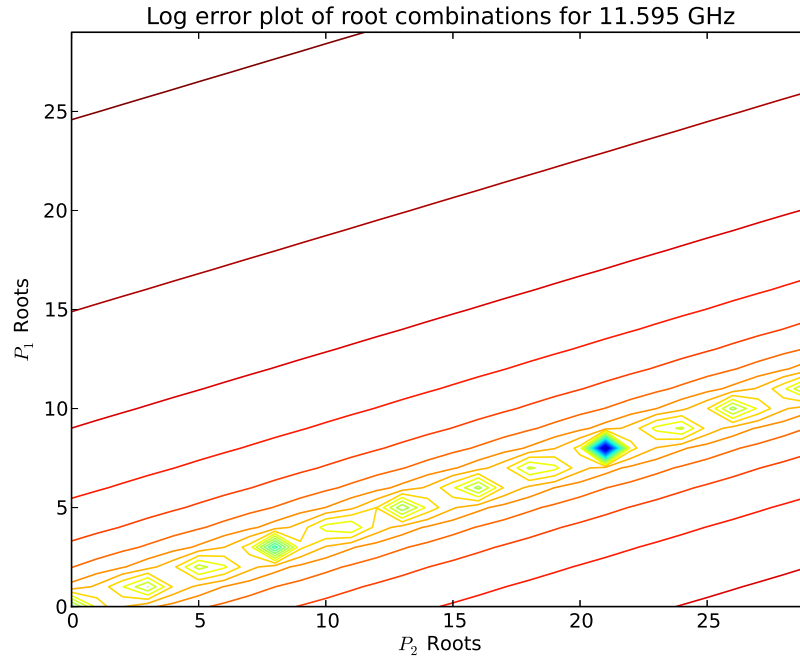


Figure 3.10: Root combination minimum error plot at 11.6 GHz. This plot solves the left-hand side of Eq. 3.3.5.

material, d_1 and d_2 are the lengths of the respective materials and $ArgZ_1$ and $ArgZ_2$ are the measured phase of the two measurements. Solving this equation requires an iterative approach through the p_1 and p_2 variables and finding the minimum value. This is successfully done and shown using a two-dimensional contour plot seen in Fig. 3.10. This is not the only root combination found throughout the frequency range of the measurement. However, this is the only combination that satisfies the uniqueness criterion where $\frac{P_1}{P_2}$ must be a rational number. Table 3.5 shows three root combinations with the least error and demonstrates how $P_1 = 8$ and $P_2 = 21$ is the only unique combination. With the most likely roots now clear, the final parameter extraction is possible. For 2D contour plots of the other possible root combinations, as well as, the code used for determining the minimum error roots see Appendix B.5.

$$\eta^2 = - \left[\left(\frac{c}{\omega d} \ln(e^{\gamma d}) \right) \right]^2 \quad (3.3.3)$$

$$\ln(e^{\gamma d}) = \ln|e^{\gamma d}| + j(\phi + 2\pi n) \quad (3.3.4)$$

$$p_2 d_1 - p_1 d_2 = \frac{d_2 ArgZ_1 - d_1 ArgZ_2}{2\pi} \quad (3.3.5)$$

Combination	P_1	P_2	$\frac{P_1}{P_2}$	Unique
11.6 GHz	8	21	0.380952...	Yes
12.4 GHz	10	25	0.4	No
18 GHz	7	18	0.388888.	No

Table 3.5: Lowest error root combinations uniqueness table

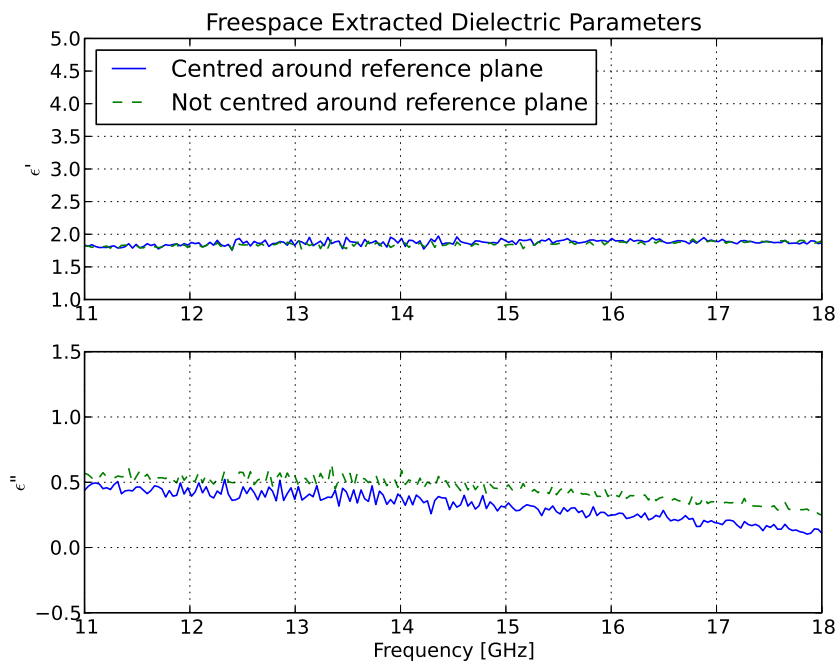


Figure 3.11: Free-space extracted parameters of the foam. A slight ripple is present in the extraction due to calibration simplifications.

The final extracted parameters for the free-space system can be seen in Fig. 3.11. The average value is consistent to that of the waveguide measurements. However, some noise is associated with the measurement because of practical simplifications in the calibration. This noise can be removed by time-gating all of the calibration and material measurements.

3.4 Coaxial measurement

The available coaxial measurements system, which functions between 2 and 6 GHz, would have aided in finding the currently missing 4 to 7 GHz data range. This apparatus consists of a flat-open-ended coaxial interface, otherwise known as a probe. This interface is placed against the MUT. However, the mechanical nature of the probe measurement restricts itself to a small measurement area. The latter is about the size of a (Sub Miniature version A) SMA connector.

This approaches the size of the micro-structure in the foam and, as a result, affects the measurements. The foam is constructed out of small carbon-coated foam balls with air in between. When measuring in a small area, the air part starts to dominate the measurement. Using a larger diameter connector, such as an N-type, will decrease the highest measurement frequency making it irrelevant. Therefore, the final measurements will not make use of the coaxial probe; however, the measurements can still be viewed in Appendix C.

3.5 Dielectric fit

With most of the 18 GHz band characterised, a fitting scheme is used to assume values for the undefined regions, as well as to simplify the result into an analytical curve. The scheme will use a random 80 % of the points to fit a polynomial curve and use the remaining 20 % to test the error of the fit. Using this scheme, the best order of the polynomial fit can be applied while not over or under-fitting the data, see Eq. 3.5.1.

$$\epsilon^x = C_0 + C_1x^1 + C_2x^2 \dots C_{n-1}x^{n-1} + C_nx^n \quad (3.5.1)$$

The raw data from the three measurements can be seen in Fig. 3.12. Here it can be seen that the dielectric has a slight decreasing trend. However, the overall trend is rather constant, and as a result, the unknown regions will also be assumed to be relatively constant.

Using the discussed scheme, the following error scores were compiled for the different orders as seen in Table 3.6. There is a definite decrease in error gain after a 6th-order fit. Therefore, to avoid over-fitting, a 6th-order polynomial was chosen. The constants for the polynomial, also seen plotted in fig. 3.13, can be found in Eq. 3.5.2. The error terms were calculated as the average of the absolute difference between the test-set data and the fitted trend. To avoid the polynomial from diverging after 18 GHz, where the measured data ends, a log fit was used to extend the fitting data up to 20 GHz. The latter log fit is described by Eq. 3.5.3 and can also be seen plotted in Fig. 3.13. The form of the log fit might be easier to use. However, some CEM simulation packages only allow for a polynomial entry. The python code used for fitting the polynomial curves can be seen in Appendix B.6.

Order [n]	0	1	2	3	4	5	6	7	8	9
Error	47.0	20.1	18.6	17.8	16.6	12.8	10.7	10.7	10.5	10.3

Table 3.6: Amount of linear test error for each order fit.

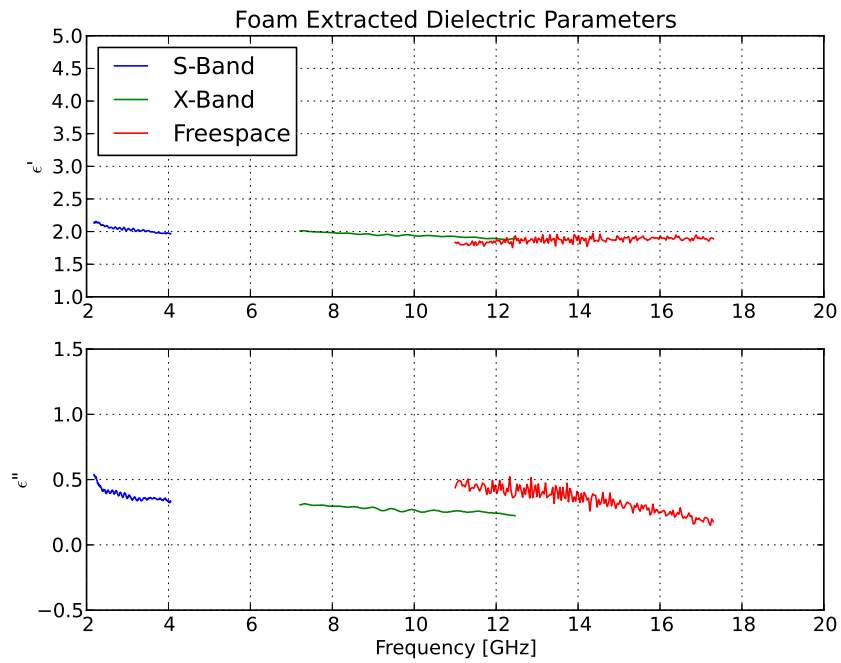


Figure 3.12: Dielectric data over all of the three measurement campaigns.

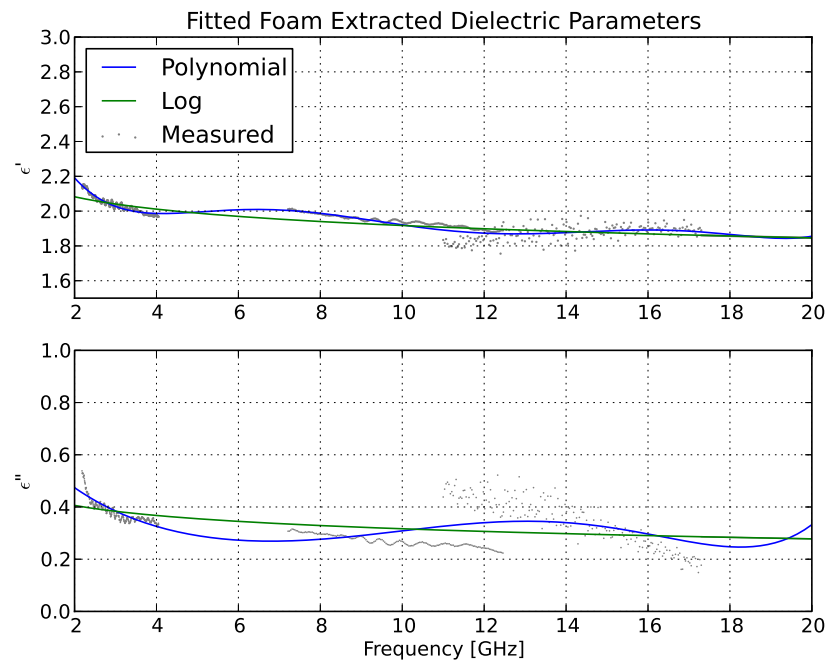


Figure 3.13: Fitted polynomial and log trend.

$$\begin{aligned}
\epsilon_{r(real)} &= 1.72E^{-6} - 1.22E^{-4}x^1 + 3.36E^{-3}x^2 - 4.61E^{-2}x^3 \\
&\quad + 3.24E^{-1}x^4 - 1.10x^5 + 3.42x^6 \\
\epsilon_{r(imaginary)} &= -3.47E^{-7} + 1.82E^{-5}x^1 - 3.84E^{-4}x^2 + 4.73E^{-3}x^3 \\
&\quad - 4.2E^{-2}x^4 + 2.3E^{-1}x^5 + -7E^{-1}x^6
\end{aligned} \tag{3.5.2}$$

$$\begin{aligned}
\epsilon_{r(real)} &= -1.026E^{-1} \log_{10} x + 2.153 \\
\epsilon_{r(imaginary)} &= 5.562E^{-2} \log_{10} x - 4.445E^{-1}
\end{aligned} \tag{3.5.3}$$

3.6 Conclusion

In this chapter, multiple existing techniques were used to extract the dielectric parameters of a carbon-impregnated foam. Using the array of systems, a broadband set of data was captured. A relatively steady trend for the dielectric constant was found through the measured band. The coaxial measurements were not used in the final dataset due to the complications brought on by the micro-structure of the foam block. In the end, the final data set was fitted to a sixth-order polynomial as well as a logarithmic line, which infers the remaining unmeasured sections of the band. The characterised foam block can now be added to the computational models that, in the next chapter, will be compared to measured scale models. This chapter did not attempt to produce original work. However, the content in this chapter needed to be mastered in order to proceed with the project.

Chapter 4

Computational Modelling and Scale Measurements

4.1 Introduction

Determining the effects of the berm on terrestrial propagation is essential to the SKA-SA effort. However, real-world measurement campaigns are expensive and time-consuming endeavours. Consequently, it would be of benefit to obtain an accurate computational model to investigate the shielding features of the berm. Therefore, a simplified scaled down model will be carefully measured and used to validate its computational representation. Additionally, an established model makes it possible to investigate permutations of the same soil structure.

The dimensions of the scale model were determined in accordance with the dielectric block characterised in the previous chapter. The scale (1:43) was chosen as the ratio between the shortest axis of the dielectric slab to the height of the real-world berm. With this ratio, the frequency scaling can be seen in Table 4.1. The highest frequency measurable in the high-frequency laboratory of Stellenbosch University is 20 GHz. Because of this, the highest scaled measurable frequency is only 465 MHz. Be that as it may, if the computational model is verified up to this frequency, it can be used to investigate higher frequencies beyond the reach of the laboratory measurements. The dimensions of the dielectric block can be seen in Table 4.2.

This chapter will start with the scale model and how it was measured over

Full size	Scale model
50 MHz	2.15 GHz
250 MHz	10.75 GHz
450 MHz	19.35 GHz

Table 4.1: Frequencies on the scaled model compared to the full-size berm.

Width	880 mm
Length	1160 mm
Height	280 mm

Table 4.2: Dimensions of the characterised foam block.

the 20 GHz bandwidth. Thereafter, the setup and validation of the computational model will be discussed. It will be seen that the latter model, even at the measured frequencies, uses a substantial amount of resources. Hence, an infinite approximation is used, which in the case of MoM code reduces the resource dependency dramatically. Similarly, this infinite approximation is once again validated using a quasi-infinite ground plane technique. This infinite approximation made it possible to simulate the entire berm using hybrid MoM and GO techniques. This allowed investigations into the shielding effectiveness of the berm. Furthermore, the effect of a metallic shed next to the KAPB could also be included in simulations. As a test, this computational model will be compared to a full-scale berm measurement in a later part of this work.

4.2 Measurement of the scale model

In the scale model measurements, the objective is to create a computational model that accurately describes the measurement. Therefore, the scale model does not have to replicate the berm itself, but rather represent a simplified version thereof. To start, a plate with the dimensions seen in Fig. 4.1 was used as a ground plane. On each end of this plate, there is a small-monopole antenna that is fed from the bottom. These antennas will be used as the principle method of comparison to the computational model. Here S_{11} and S_{21} will be used, where each antenna (Fig.4.2) represents its own port. The dielectric will then be placed in between these antennas to simulate an obstruction such as a berm.

Care was taken to measure the model as accurately as possible. The measurements were done inside an anechoic chamber, with well-known calibration standards. Additionally, isolation, power slopes, as well as averaging on the 201 data points were used to ensure a reasonable amount of sensitivity. While the chamber helped to remove environmental effects from the measurement, it involved the use of long cables which, at these frequencies, were causing the measurement to lose sensitivity. It was finally decided to move the system out of the anechoic chamber and shorten the cables, which resulted in a required more-sensitive analysis. Consequently, the measurement had to be time-gated to remove any effect from the surrounding laboratory. The setup of the measurement inside the chamber can be seen in Fig. 4.3.

The measured S-parameters of the system outside the chamber, for the case where no dielectric is present, can be seen in Fig. 4.4. The antenna, being a

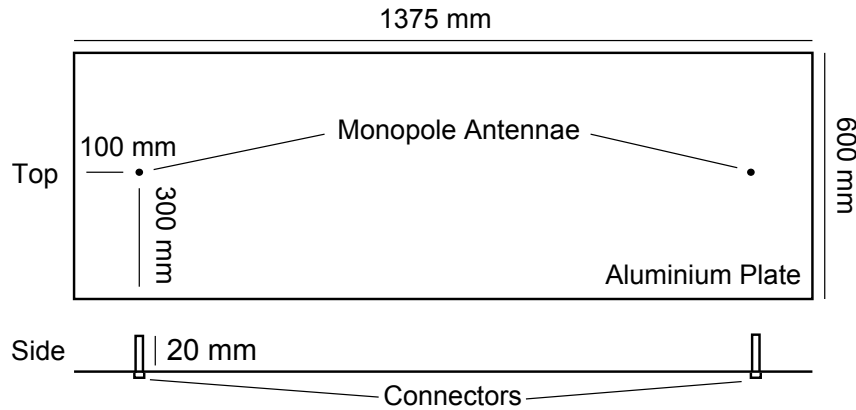


Figure 4.1: Diagram of the metallic plate used for scale measurements. The antennae are located at each end of the plate, with the aluminium mounted on a wooden base for extra mechanical support.



Figure 4.2: Diagram (left) and photo (right) of the fat monopole antenna, mounted on the metallic plate, with its feeding connector on the bottom side of the structure. The antenna was machined out of brass material.

simple monopole antenna, has definite resonant points seen in the S_{11} plot. The latter causes the small coupling at about 8 GHz. However, because this will be used to validate the computational model it will be sufficient. A wideband mono-conical antenna was developed to ensure better sensitivity across the band. Due to time constraints this was never compared to a computational model. The S_{21} graph clearly shows a decreasing trend at the resonant sections of the antennas, which is expected for partly free-space propagation. In the next section, the computational model will be discussed and compared to measured data.



Figure 4.3: Photo of the scale-model measurement setup inside the anechoic room.

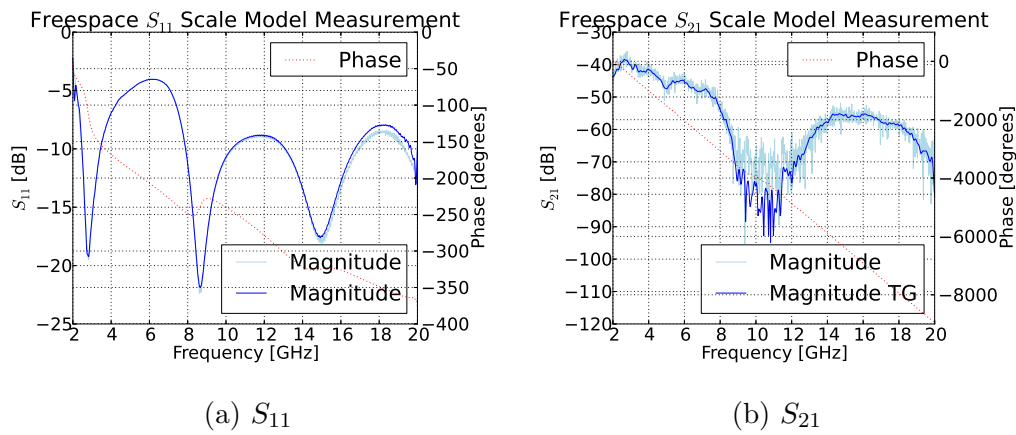


Figure 4.4: Scale model, without dielectric, measured outside of the anechoic chamber with a 10 ns time-gate.

4.3 Computational modelling using an FDTD code

4.3.1 Simulation setup

Using CST MWS[®], the measured model could be simulated using an FDTD code. The entire plate, antennas and dielectric were modelled as closely as possible. This included the correct material parameters as well as the wooden-support base underneath. Care was taken to ensure that detailed features such as the antenna feeds were accurately meshed. This included extra mesh elements for dielectric sections, such as Teflon and the dielectric block under test (see Fig 4.5). An H-field symmetry plane through both antennas was added to the simulation parameters, decreasing overall computation time. Because this is a free-space problem, boundaries were set to open. FDTD codes require boundary parameters to be carefully defined to limit erroneous reflections. Therefore, an iterative approach was used to check for convergence of simulated results. Here the estimated reflection was decreased until convergence was deemed satisfactory. A similar approach was used to set the size of the bounding box around the model. Finally, the simulation time needed to be set to ensure the energy at port 2 has sufficiently subsided.

4.3.2 Simulation comparison

With confidence in the simulation parameters, it is possible to simulate the measurement scenarios. These models in Fig. 4.6 were simulated using a third generation i7 processor with 16 GB RAM. This allowed the entire simulation to complete in 24 hours. However, the dielectric simulation was limited to

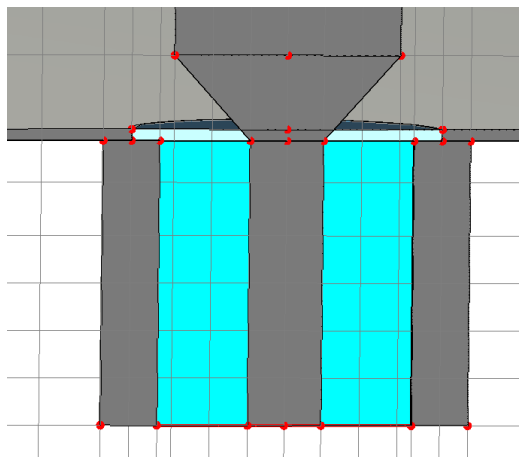


Figure 4.5: Antenna feed mesh in CST MWS[®]. Here it can be seen how all of the detailed features have been meshed. Additionally the connector dielectric section has been given extra mesh cells to account for its shorter wavelength.

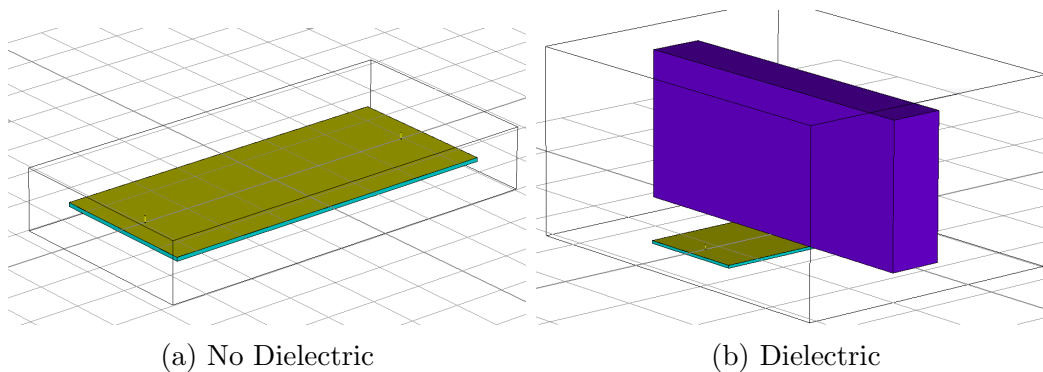


Figure 4.6: Computational models constructed in CST MWS[®]. A simulation with and without the dielectric was used for the measurement comparison. The dielectric and the non-dielectric measurement contained respectively 227 and 190 million mesh cells.

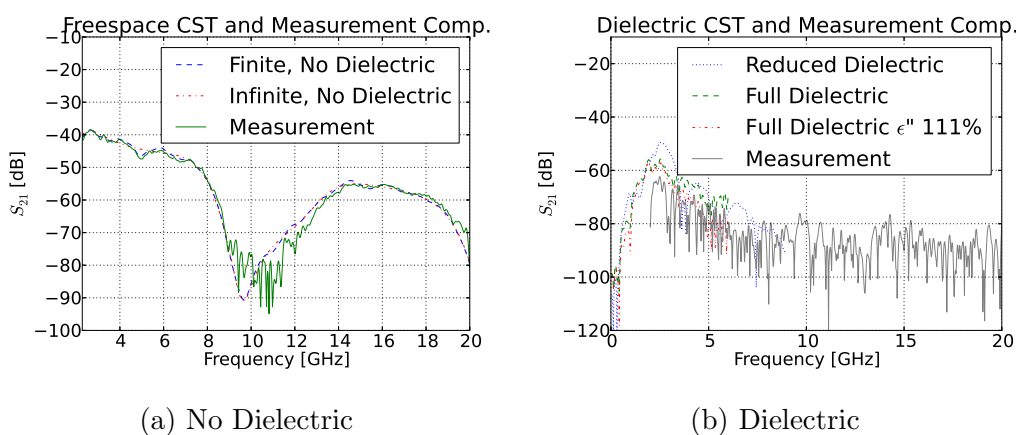


Figure 4.7: Both measurement and simulated scenarios, without the dielectric (a), and with the dielectric (b), are compared. A more efficient infinite approximation is also shown in (a).

only 8 GHz due to resource restrictions. The simulation compared to the computation results can be seen in Fig. 4.7. Also in this figure, a simulation using an infinite plate is demonstrated. The simulation is similar to that of the finite plate except that the vertical x and y boundaries are set against the plate edges. This helps to decrease the size of the meshed volume, subsequently decreasing simulation time. These comparisons are all compared using the magnitude of S_{21} between the two antennas.

It can be seen that in the case where no-dielectric is present, the agreement is good. It should also be noted that some noise is visible at the lower power levels. It is interesting to see that the infinite approximation follows the finite simulation closely with slightly less ripple. This will be further discussed in the next section. In the case where the dielectric is present sensitivity becomes

a problem. Nevertheless, there is good agreement with a slight overestimation by the simulation. As a test, ϵ'' was increased to 111% of its actual value. With this factor, the agreement shows an overall closer trend. Although this factor gives a better agreement, it will not be used in further studies. To test another simplification, the dielectric was shortened in the simulation to the edges of the plate. This significantly decreased the simulation cost down to 123 million mesh cells. The overall trend of this simulation again closely followed the measurement but showed a significant difference in its smaller details. These comparisons gave sufficient confidence in the CST MWS® models to be used for further practical purposes.

During the comparison, it became evident that the simulation is electrically very large. Meaning a high degree of discretisation is needed for the relevant frequencies. In the case where the dielectric is included in the simulation, the frequency was limited to 9 GHz. This is not ideal for the purposes of the investigation. The next section will focus on simplifying the simulation by using infinite conditions.

4.4 Quasi-infinite ground plane

Fig. 4.6 illustrated how infinite conditions can help decrease simulation time. These infinite conditions are handled even more efficiently using a MoM code. In a MoM simulation, a half-space configuration will limit the meshed elements to only the dielectric block and antenna. This eliminates the need to mesh the expensive ground plane. To use this approximation, a similar process is followed where a measured model is compared to its simulation. However, an infinite ground plane does not exist. Therefore, a quasi-infinite ground plane is constructed. This is done by using a special arrangement of plate edges and time-gating.

The difference between the infinite and finite plates is the diffracted energy from the edges and reflected currents on the plate. The reflected currents are minimised by using time gating after the arrival of the incident wave. This however, does not work for the diffracted energy, which arrives very close in time to the incident wave. To minimise this effect, the four edges are optimised into a serrated form using a simplified GO technique. This method is used to optimise the reflector in compact ranges. The optimisation involves discretising each side of the plate into slanted angles, forming serrations. These angles are then set individually to limit the diffracted rays from entering a specified zone. In this case the zone is defined as the whole plate, meaning all diffracted rays will be directed away from the experiment.

A simple optimisation code has been written assuming a 2D environment. This implies that the launch and diffract angles of the rays do not take into account the height of the antenna. To keep the problem simple, the plate edges itself were discretised into points of incidence. These points of incidence

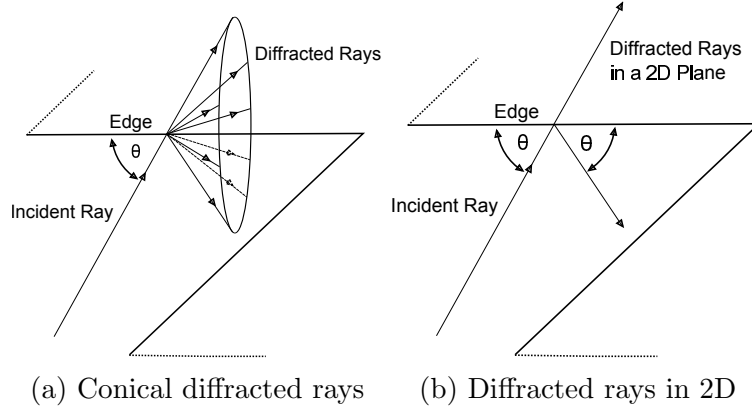


Figure 4.8: Illustrating how rays are diffracted on a thin PEC serration with an oblique angle. To illustrate the optimisation case, a 2D version is also shown.

each has its own ray travelling from the source. This eliminated the need for complex collision detection on the edges and greatly simplified the problem. Because the rays in this case are not launched in an isotropic manner, field calculations would not be valid. Nevertheless, the point of this simulation is to minimise the number of rays entering a specified zone. Therefore, phase and amplitude can be ignored. Only the paths of the rays are of significance. Each interaction with an edge is handled as a cone diffraction around a thin PEC plate (see Fig. 4.8). Because this is a 2D problem only two of these conical rays are visible. One ray is reflected with the same incident angle while the other ray has the same path as the incident ray [66].

The travelling equations used in Geometrical Theory of Diffraction (GTD) to calculate the incident and diffracted fields can be seen in Eq. 4.4.1 [67]. The simplified form used in the optimisation ignores phase, removes attenuation due to the wavefront shape and ignores the amplitude effect of the diffraction. These assumptions only leave the direction of the rays themselves. Therefore, all of the amplitudes can be assumed equal ($E_i(s) = E_d(s) = 1$).

$$\begin{aligned}
 E_i(s) &= E_0(0)e^{-jk\psi(0)}\sqrt{\frac{\rho_1\rho_2}{(\rho_1+s)(\rho_2+s)}}e^{-jks} \\
 E_d(s) &= E_i(Q_D)De^{-jk\psi(0)}\sqrt{\frac{\rho}{s(\rho^d+s)}}e^{-jks}
 \end{aligned}
 \tag{4.4.1}$$

where

- E_i and E_d is the incident and diffracted fields
- k is the wave number
- $\psi(0)$ is the initial phase
- ρ_1 and ρ_2 describe the expansion of the rays as a spherical wavefront

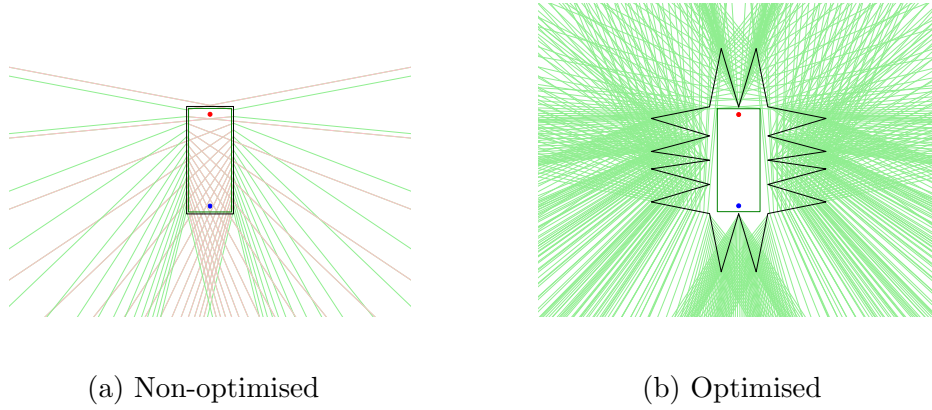


Figure 4.9: Diffracted ray comparison for an optimised and non-optimised plate, using a simplified GTD approximation. The dot at the top of the plate represents the transmitting antenna while the surrounding square indicates the zone that is optimised to be free of diffractions. In the non-optimised case, red rays indicate rays that have passed through this zone.

- s is the distance travelled along the ray
- ρ^d has to do with the geometry of the local diffracted rays
- D is the diffraction coefficient
- Q_D is the point of diffraction

Fig 4.9 shows the difference between the diffracted rays for the optimised and non-optimised cases. This optimisation only takes into account the primary diffracted rays, ignoring their secondary diffractions. The rays are calculated using simplified GTD. The optimisation algorithm used for the serrated plate can be found in Appendix D. Serrations were chosen to be at least five wavelengths, for the asymptotic GTD approximation. This method is used in compact ranges for reflected antennas [68].

As a test, the optimised plate was imported into CST MWS[®] to be compared to its infinite case. Fig. 4.10a shows a simulation between the infinite and quasi-infinite plate to up to 2 GHz. It is interesting how the serrated edge closely follows that of the infinite plate with only a slight offset. It can be seen how the infinite-plate simulation is smooth as opposed to the oscillations seen in the finite plate. This oscillation is due to the mismatched-reflected currents on the plate. The mismatched waves only arrive at the receiver antenna after 9 ns, 4 ns after the direct pulse (see Fig. 4.10b). These reflected pulses can be reduced by trying to match the end of the plate to free space. However, to save time, these mismatched waves can be time-gated for a direct comparison.

After optimisation, the quasi-infinite experiment was constructed using the original ground plane while adding the extra serrations. The serrations were

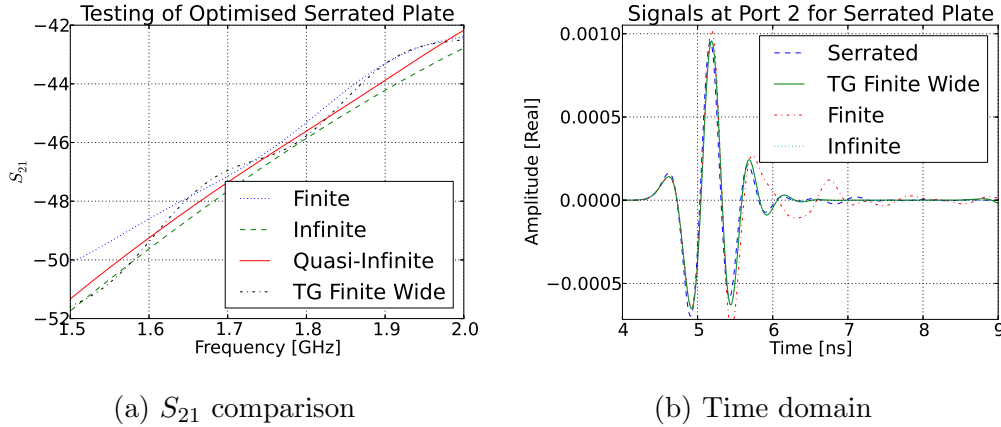


Figure 4.10: Testing of the serrated plate compared to the infinite case using CST MWS[®]. In (a) the S_{21} coupling is compared showing good agreement between the time-gated-serrated measurement and the actual infinite case. The time domain received wave can be seen in (b), showing the direct and reflected waves. TG - Time-gated.

attached using copper tape supported by wooden stilts. After that measurements were conducted in the same manner as in Section 4.2. The construction and measurement can be seen in Fig. 4.11. It can be seen that the quasi-infinite measurement compares well with the infinite CST MWS[®] simulation. However when calculating the RMS error between the quasi-infinite and finite to the infinite case, the finite measurement has a lower error. Nevertheless, the overall trend of the quasi-infinite measurement follows the infinite case better with a 2 dB offset causing the overall error. The quasi-infinite case successfully helps to reduce the oscillations present in the finite measurement. With this comparison, further CEM work can be done using more efficient infinite approximations.

In CST MWS[®], the infinite conditions only decrease the bounding box in the X and Y direction by 200 mm. Because this whole volume still needs to be meshed, the gain in efficiency is not enough to make 20 GHz simulations practical. However, by using a MoM code a significant decrease in simulation time can be achieved. MoM only discretises the modelled surfaces and not the entire volume. Therefore, by using an infinite-PEC half-space approximation for the plate, the meshed surfaces significantly decrease. This becomes more important when field values, far out of the bounding box, need to be simulated.

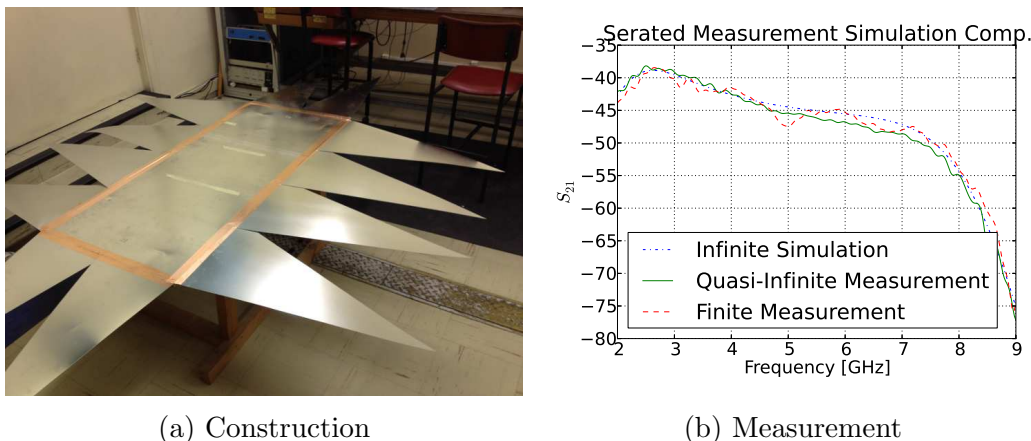


Figure 4.11: Constructed serrated plate with its non-dielectric measurement. Here the serrated time-gated measurement is compared to a full infinite plate CST MWS[®] simulation.

4.5 Computational modelling using a MoM code

Using a PEC half-space in a MoM code will remove the need for meshing the finite plate. This means, for the current measurement configuration, only the antennas and dielectric need to be meshed. This will also aid in more general simulations. In these simulations, the antennas can be replaced by defined sources and the dielectric by a representation of a berm. In this section, hybrid methods are investigated to dramatically reduce simulation time.

Fig. 4.12a shows a FEKO[®] simulation using a PEC half-space to simulate the no-dielectric case. This is compared to the benchmark infinite-simulations in CST MWS[®] as a control Fig. 4.12b. The infinite ground-plane simulation using FEKO[®] took less than 30 minutes on the same machine used for the CST MWS[®] simulations. Although the MoM simulation follows the same trend as the FDTD code, a 5 dB difference exists in some areas. In these areas, the MoM code underestimates the coupling. This is most likely due to the difference in feed configuration. In CST MWS[®], the feed is modelled as close as possible to the SMA interface on the bottom of the plate (see Fig. 4.5). Because of the halfspace approximation, the same is not possible for the FEKO[®] simulation. In the FEKO[®] simulation, the feed is defined as an edge port between the base of the antenna and the halfspace. This will affect how the electric field forms at the base of the antenna as well as the reference phase of the simulation. For practical simulations, the antennas will be replaced by defined sources where this would not be a problem. However, for now simulations will still use this familiar antenna coupling configuration. It will shortly become evident that it is also possible to exploit image theory [52]. This means that the halfspace approximation can be removed by replacing the

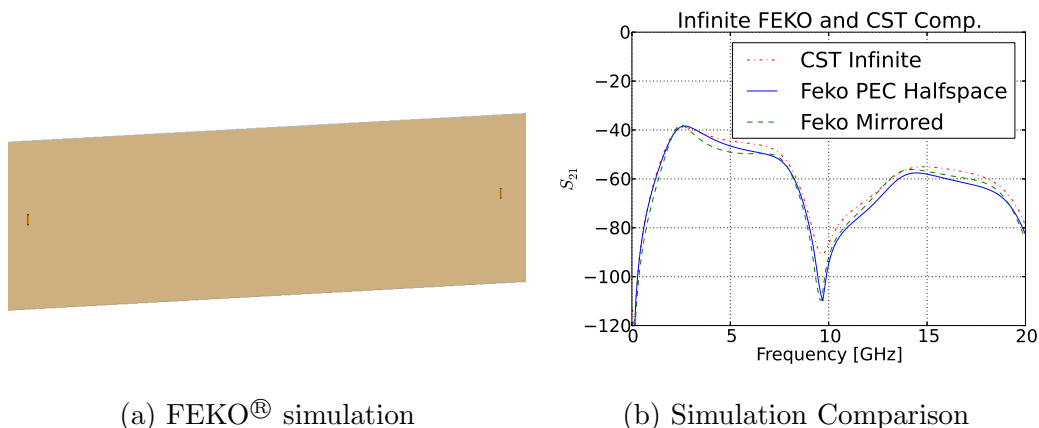


Figure 4.12: FEKO[®] PEC half-space compared to infinite CST MWS[®] simulation in (b). The two antennas can be seen on both sides of the image, on top of the infinite PEC halfspace.

monopoles with dipoles around the current ground plane. Consequently, the dielectric is then also mirrored around this surface. A non-dielectric test in this configuration is also compared in Fig. 4.12b.

With the infinite PEC halfspace, a dielectric can be added to test its simulation time against the MoM code. As soon as the dielectric is added, simulation time increases to the point where a frequency of 20 GHz becomes unpractical again. Even meshing becomes problematic on a desktop computer above 2 GHz. Consequently, hybrid methods need to be used. These hybrid methods combine the classical MoM with GO, called Ray Launching Geometrical Optics (RL-GO). In this hybrid simulation, the MoM will handle small-detailed structures while the asymptotic method handles the larger well approximated structures. These methods are well suited for this problem because of the electrically large dielectric and the small antennas. The problem here however is that an infinite halfspace cannot be used in conjunction with RL-GO. Therefore, the infinite-PEC halfspace is replaced with the previously mentioned mirror principle. In this case, the dielectric could be solved using GO. At the start, the default ray launching angles were too coarse, this had to be manually changed in the simulation settings. A GO only method would also be possible by replacing antennas with spherical-mode sources. The comparison between the simulation and measurement of the dielectric can be seen in Fig. 4.13b. Here the measured dielectric on the quasi-infinite plate is compared to the mirrored simulation. This figure shows a simulation where the antennas were approximated as line monopoles. The simulation itself finished in the order of minutes. The comparison can be seen to underestimate the measurement at some frequencies. It should be noted that the feeds cannot be accurately modelled in this configuration and is, therefore, partly attributed to the deviation. Also, the FEKO[®] simulation does not take into account the

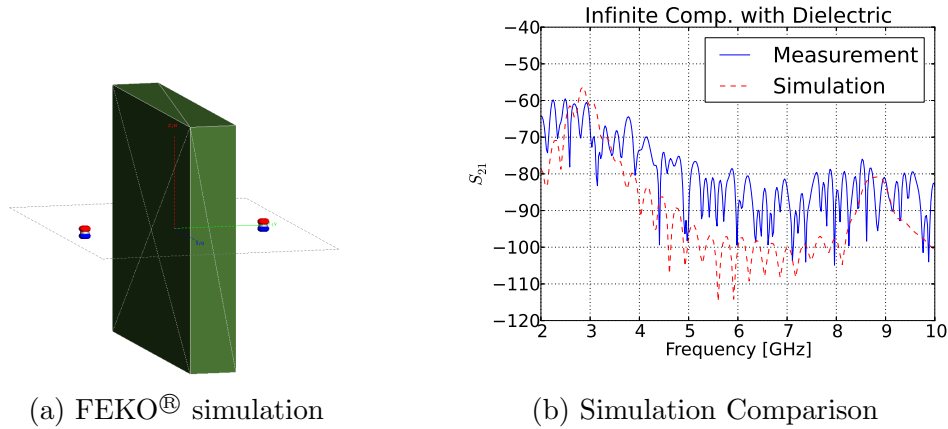


Figure 4.13: FEKO[®] mirrored simulation including dielectric. Results here are compared to quasi-infinite measurements.

slight rounding of the edges which would significantly decrease diffraction. It should be noted that GO does not directly account for diffraction (GO only accounts for reflection and refraction). However, some relation between RL-GO and physical optics (PO) exists in that elementary Huygens sources are generated on surfaces. Surface currents on the rest of the structure are consequently induced by radiation from these sources [69][70]. This approximate PO-like behaviour makes it important to verify the simulation against full-scale real-world measurements. These real-world comparisons can be found in Sec. 6.4.2 where a Multi-copter vehicle is used to do the measurements at this large scale.

4.6 Full berm simulation

With the mirrored configuration in the previous section, it is possible to start modelling the full-size berm. Just like the dielectric block, the berm is mirrored around the ground plane. This simulation assumes a flat-PEC earth due to the nature of the approximation. The antenna for this simulation was chosen as a resonant monopole (260 MHz) on the ground plane. This frequency is one of the lowest frequencies measurable with the Multi-copter system discussed in the next chapter. The current simulation configuration is easily capable of much higher frequencies within practical time-scales. The FEKO[®] model used to simulate the full berm along with the resonant monopole can be seen in Fig. 4.14. The material parameters for the berm was set to the Karoo-soil parameters measured in Otto [4].

The simulation was configured to measure a 2D plane through the middle of the berm, starting 20 m in front and extending to 200 m behind. The results of this simulation can be seen in Fig. 4.15. To ensure the correct functioning of

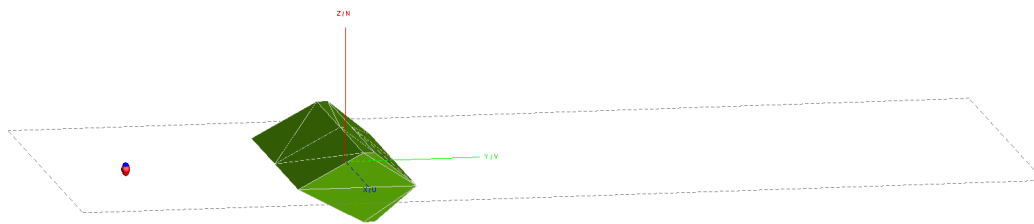


Figure 4.14: Full berm simulation with antenna 50 m away. The berm and antenna were both mirrored around the ground plane to make RL-GO in FEKO® possible.

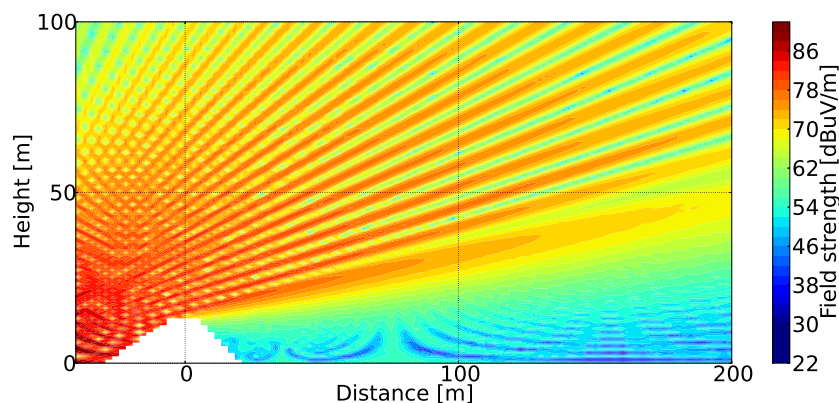


Figure 4.15: Simulation of a vertical plane, showing the E-field through the centre of the berm at 260 MHz. The source used in this simulation is the simple resonant monopole on the ground. The solution is not valid inside dielectric volumes. Therefore, the area that contained the berm is left blank.

the RL-GO, the ray-tracing angle step needs to be set to at least 0.095 degrees for this frequency.

4.6.1 Simulating berm shielding

Using this configuration, the shielding of the berm can be extracted. This is done by comparing the simulation to a configuration where the berm is removed. Subtracting the berm simulation from the free-space data will leave the shielding contribution of the berm only. A shielding graph generated in this manner can be seen in Fig. 4.17. Take note that the shielding at any point will be a function of the transmitting antenna position. From here it is clear the berm does have shielding capabilities. The diffraction over the berm into the shadowed area is clearly visible as 2D oscillations. These diffractions lower the shielding in some areas to below 15 dB. Nevertheless, the **shielding in this region is always positive**. This means that the diffracted rays do not

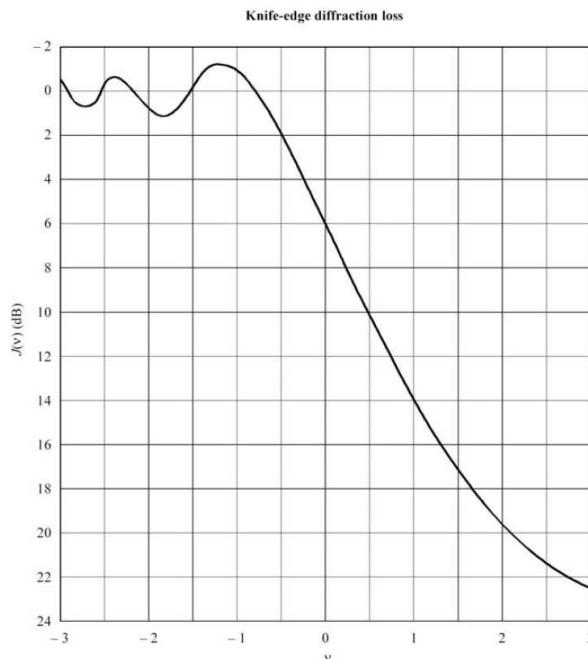


Figure 4.16: Fresnel integral as a function of v . This graph was taken from Seybold [22].

increase the noise in the shielded region.

This is also apparent when analysing the diffraction equations. These equations are reliant on a Fresnel integral in the form seen in Fig. 4.16. In this case v represents the geometry of the transmitter, receiver and diffracting obstacle. Assuming the case where both the transmitter and receiver is below the height of the obstacle v will always be positive. This holds for all possible transmitter and receiver locations. This means, with reference to Fig. 4.16, that there will always be a positive loss. However, this might not hold for the case where either the transmitter or receiver is higher than the obstacle. These findings are important in the context of the Karoo site base as discussion amongst site engineers expressed concern that the berm could in some cases worsen fields in the shadow region. This study shows that the berm will always have advantageous properties in terms of shielding effectiveness. This is a significant finding for all berms on radio astronomy sites.

4.6.1.1 Shielding effectiveness to an existing MeerKAT receiver location

The shielding effectiveness of the berm can be simulated for MeerKAT receiver M60 (see Fig. 6.15) using the model developed up to this point. This receiver is located about 5 km on the far side of the berm from the KAPB. To expedite the investigation, a reciprocal approach is used where the source is placed at the location of the receiver (15 m above ground). This allows the

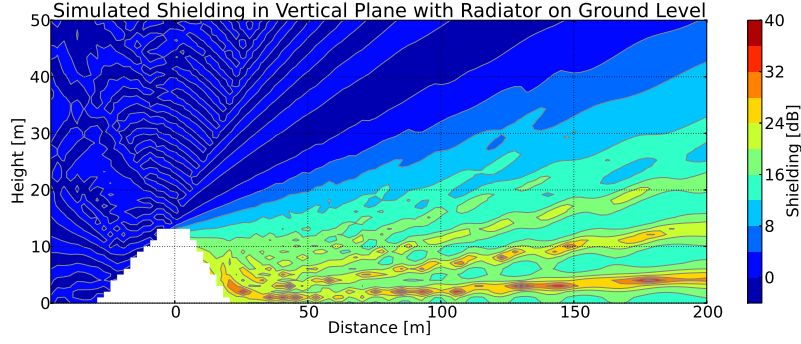


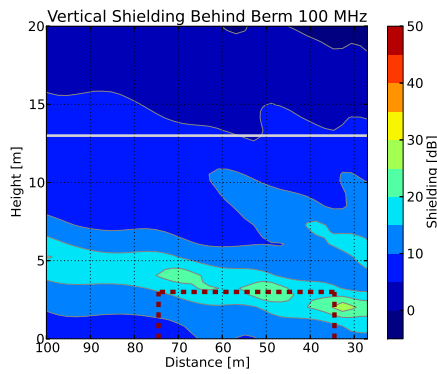
Figure 4.17: Simulated shielding, with a monopole on the ground, 50 m in front of the berm at 260 MHz.

simulation of an array of points in the KAPB area with a single simulation run (see Fig. 4.19). A 100 m by 100 m horizontal area at 2 m high along with a 20 m by 100 m vertical plane was chosen to be simulated. This area will give a good indication of the shielding profile created by the berm in the KAPB area. It should be noted that this is an ideal simulation that does not include effects of buildings and other terrestrial structures. Fig. 4.18 shows the simulated shielding simulations for both the horizontal and vertical planes at three distinct frequencies. These frequencies, 100, 220 and 900 MHz were chosen roughly to represent the berm height in terms of 4, 10 and 40 wavelengths respectively. These simulations illustrate how shielding undergoes a dramatic decrease below 220 MHz (10 wavelengths). Above 10 wavelengths, the shielding remains relatively constant in the marked out area representing the KAPB. On each of the simulations, a dashed outline highlights the location of the KAPB. To evaluate the performance of the berm at different frequencies a figure of merit (FOM) was needed. The FOM was chosen to be the average and minimum shielding in the KAPB volume to maintain relevancy with practical objectives. Formal definitions for the average shielding effectiveness (ASE) and minimum shielding effectiveness (MSE) can be seen in Eq. 4.6.1 and 4.6.2.

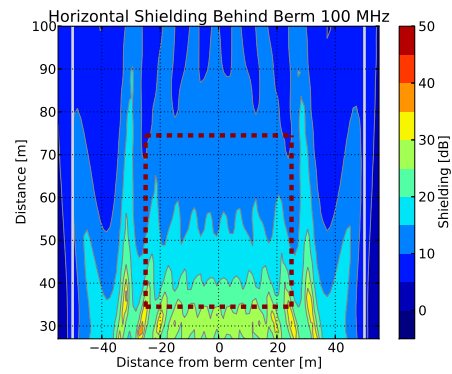
$$ASE = 10 \log \left[\frac{1}{\left(\frac{1}{P}\right) \sum_{n=1}^P |E_n|^2} \right] \quad (4.6.1)$$

$$MSE = 10 \log \left[\frac{1}{\max_{1 \leq n \leq P} |E_n|^2} \right] \quad (4.6.2)$$

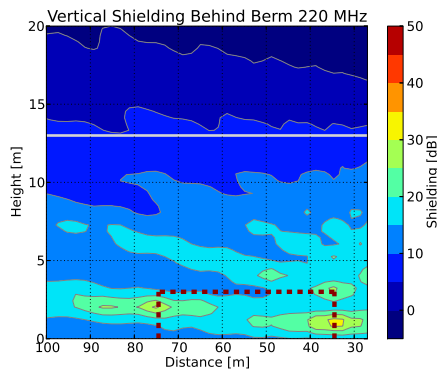
With the KAPB defined as the region of interest (RoI), the ASE and MSE were calculated for each of the three frequencies simulated in Fig. 4.18. The vertical and horizontal planes were configured as seen in Fig. 4.19. At 100 MHz, an ASE of 12 dB is found. Increasing the frequency above 220 MHz shows a dramatic gain in the ASE, which is maintained above 16 dB. This figure then rises to around 25 dB at 900 MHz with an MSE of 18 dB. Above



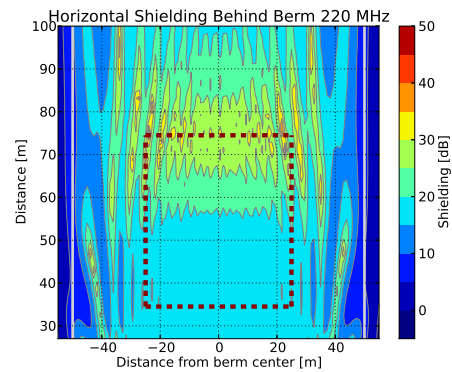
(a) 100 MHz Vertical



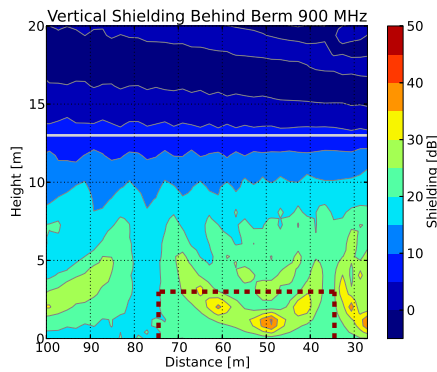
(b) 100 MHz Horizontal



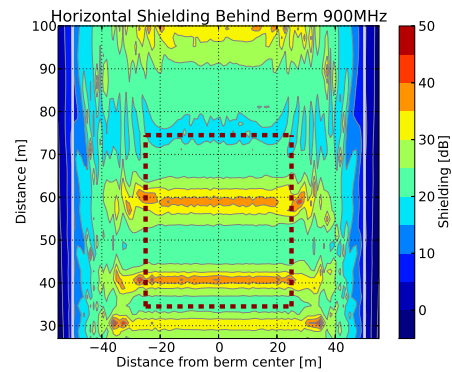
(c) 220 MHz Vertical



(d) 220 MHz Horizontal



(e) 900 MHz Vertical



(f) 900 MHz Horizontal

Figure 4.18: Shielding simulations to a nearby MeerKAT location for the berm at heights of 4, 10 and 40 wavelengths. The border from the line-of-sight to the shadowed regions are indicated by white lines. The KAPB location is roughly outlined by dashed lines in the figures. This marked volume is what is used to calculate figures of merit called ASE and MSE.

Model	FOM	100 MHz	220 MHz	900 MHz
RL-GO	ASE	12.45 dB	16.71 dB	24.63 dB
RL-GO	MSE	7.33 dB	10.91 dB	18.2 dB
2D	ASE	-	21 dB	29 dB
2D	MSE	-	14 dB	22 dB

Table 4.3: FOM results in the RoI for the 3D RL-GO and 2D boundary integral models.

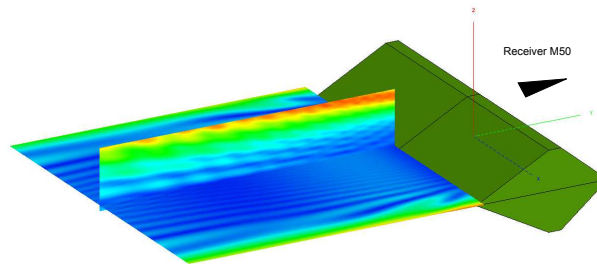


Figure 4.19: Simulated 2D planes using the mirrored berm approximation. The vertically polarised source is located at the position of receiver M60 15 m off of the ground, 5 km away. The planes illustrate the absolute electric field diffracted over and around the obstacle.

900 MHz, the simulation run-time starts to become limiting. In the event that it becomes necessary to simulate larger berms or higher frequencies, a 2D simulation model based on a single-layer boundary integral equation developed by N. S. Tezel [71] can be used. For example, a 500 MHz simulation of the berm takes around 4 hours to simulate using the 3D RL-GO method. Using the 2D boundary integral model the simulation time for the same 550 MHz scenario only takes 7 minutes. For completeness, the MSE and ASE values for both the 3D and simplified 2D model can be seen in Table. 4.3. Note that in all cases the 2D model shows around a 5 dB higher shielding. This is an important finding and elucidates the consequence of the berm edges allowing diffraction to take place around the berm into the KAPB region.

4.6.1.2 Effect of metallic structures on overall shielding

It is clear now that the berm does in fact add to the shielding effectiveness of the site. However, a shed structure (white building on the right-hand side of Fig. 6.1) has been built of which its top extends above the height of the berm. Therefore, it is likely that energy would diffract off of the metallic edges of the structure in the direction of the receiver. These diffractions would not be attenuated by the berm. As a result, the consequence of this propagation path on the overall shielding effectiveness of the KAPB site was investigated.

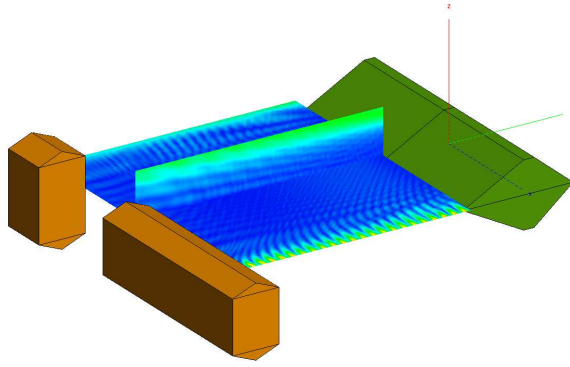


Figure 4.20: The same simulation configuration as in Fig. 4.19 with shed structures added.

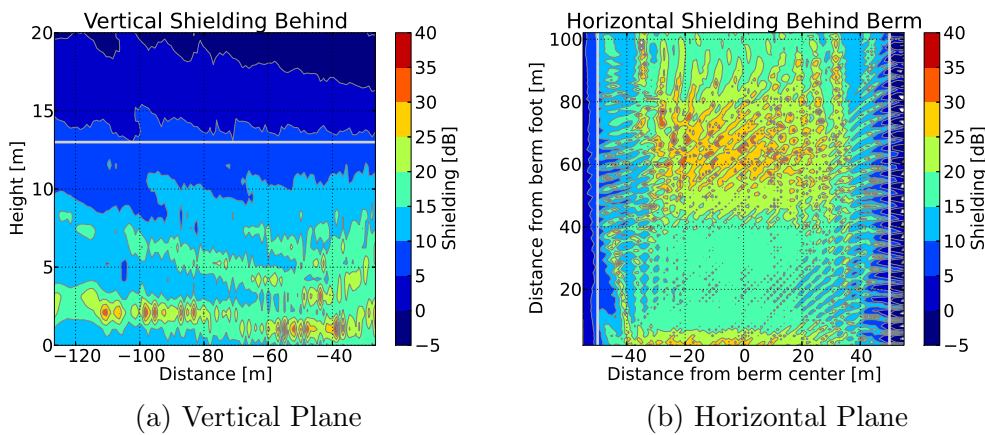


Figure 4.21: Berm shielding effectiveness in KAPB area to MeerKAT receiver M60 at 260 MHz. In this case, a large PEC shed structure has been added, causing diffraction into the shielded area.

Fig. 4.20 shows the simulation geometry of the site including the two metallic sheds. For simplicity, the shed structures have been approximated as PEC cubes with triangular roofing. Dimensions have been chosen to represent the simplified structure as accurately as possible. For the sake of comparison to the case without the shed, a frequency of 260 MHz was used again. Fig. 4.21 shows the simulated shielding plots for the case with the diffracting shed.

The diffraction into the KAPB area is clearly visible. This diffraction has a negative effect on the shielding, lowering the effectiveness of the berm in particular regions. Because these mechanisms are very much frequency dependent, it is not possible to isolate these areas. An alternative would be to shield the exposed faces which are in the LOS of MeerKAT receivers. This could be accomplished by covering these faces with scatterers or an absorbing material. These solutions could easily be extended to other permanent metallic structures on-site such as a water tower and an overhead-parking roof. In the

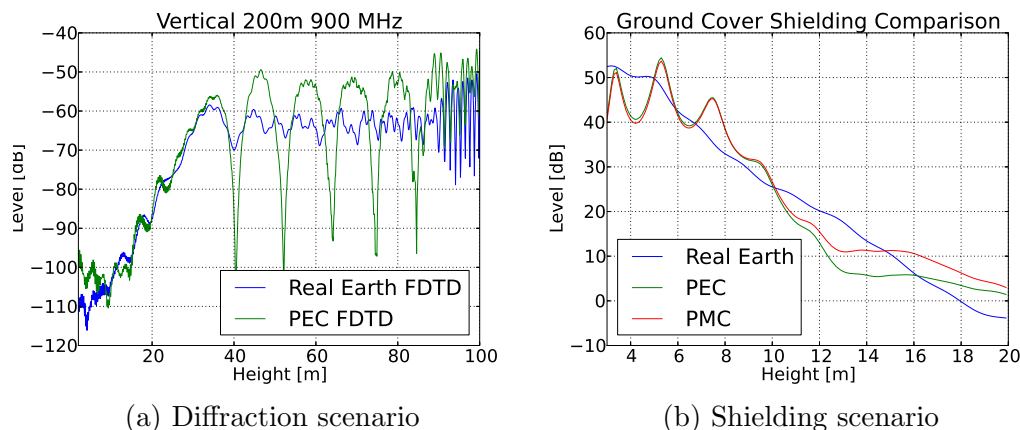


Figure 4.22: Comparison of our PEC, PMC ground cover approximating real-earth using a 2D FDTD model. Both the diffraction (non-grazing angle) and shielding (grazing angle) scenarios are tested.

case of vehicles, it is possible to shift the permanent parking area to the far side of the shed structure. These simple modifications could guarantee higher overall shielding as well as an increase in the predictability of the site shielding. Another alternative would be to add vegetation on top of the berm. Apart from stabilising the ground, this would act as an absorber for the exposed shed structure.

4.6.1.3 Testing the PEC approximation

The RL-GO 3D model simulations assume a PEC ground plane in order to achieve simulations of this electrical size in reasonable time. At VHF and UHF frequencies the reflection coefficient for a vertical polarised signal at angles which are non-grazing is close to 1 [72]. This is consistent with PEC. However, when these incident angles approach grazing, a PMC ground approximation would be more accurate. We are only interested in the shadowed region behind the berm where the KAPB is situated. Therefore, these poorly approximated reflections should be negligible. To test the approximation, a 2D FDTD model with the capacity of simulating real-earth, developed by R. G. Ilgner [73] is used.

The test process will consist of simulating the two scenarios of the RL-GO model used in this work. These scenarios consist of a diffraction simulation used in Sec. 6.4.2 and the shielding computations discussed in this chapter. Using the FDTD model, each scenario is simulated using a PEC and real-earth ground which can be seen in Fig. 4.22. In the case of the diffraction scenario, the trend between the real-earth and PEC ground traces are very similar. This is especially true in the shadowed region. The most substantial differences are caused by destructive interference from hard reflections in the non-shadowed region. The latter errors are visible as large periodic nulls. The

second scenario representing the shielding simulations is seen in Fig. 4.22b. Because of the grazing angles formed between the transmitter and receiver, an FDTD calculated PMC ground approximation is also shown for comparison. The approximate ground-cover models can be seen to oscillate around the real-earth model. For our purposes, this oscillation is counteracted by the spatial averaging function when calculating the ASE and MSE metrics. The minor improvement from using PMC in the shielding case does not justify the use of two separate grazing and non-grazing models. These comparisons give us confidence in the validity of our RL-GO 3D model, especially in the shadowed region behind the berm. Nevertheless, care is needed when using data in the non-shadowed region where periodic nulls will need to be identified as artefacts of the CEM model. The RL-GO 3D model will be tested again against full-scale measurements discussed in Chapter 6.

4.6.2 Simulating measurements

On-site measurement campaigns are costly and time-limited endeavours. Therefore, measurements need to be planned and prioritised beforehand. The computational model developed in this chapter gives the ability to simulate Multi-copter flight paths. A simple flight possible with a Multi-copter is a berm-diffraction measurement. In this measurement, the vehicle is held at a particular position while varying its height from 10 m to 100 m. Fig. 4.23 shows two such vertical diffraction simulations at 50 m and 200 m behind the berm. These simulations give a pre-emptive indication of the sensitivity that will be required for a measurement campaign. Additionally, during the campaign, these will be used as a reference for checking the reliability of a measurement.

During the first measurement campaign LS of SA offered their time to help gather data. They used their Multi-copter platform to measure a horizontal plane behind and around the berm. With this information, initial comparisons could be made to the FEKO[®] simulations. Due to the high frequency of the measurement, data behind the berm was mostly in the noise floor. Also, it was not possible at this time to de-embed the measurement into field values for comparison. Therefore, units are irrelevant in this comparison. Nevertheless, this plot still gives a sense of the propagation around the sides of the berm while highlighting the usefulness of the simulation tool.

4.6.3 Berm conductivity parameter study

Recent discussions have arisen about the claim that a 10% change in the complex dielectric soil parameters [71] would in fact have a significant effect on the measured and simulated parameters - conductivity was not measured. Equivalent dielectric conductivity is deduced from ϵ'' . This is also of interest to this dissertation as the soil parameters measured in the laboratory have changed in composition. Further to this, natural effects such as rain and seasons will

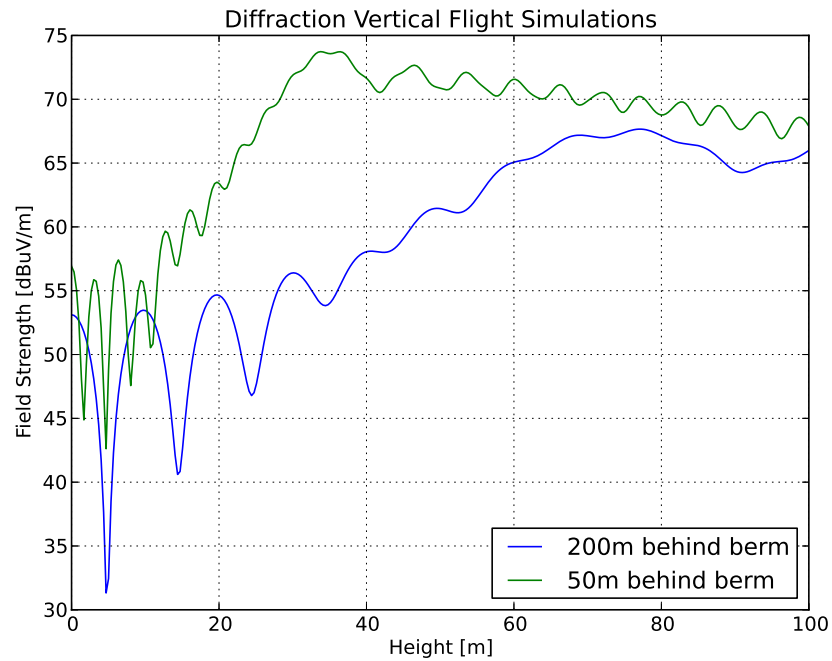
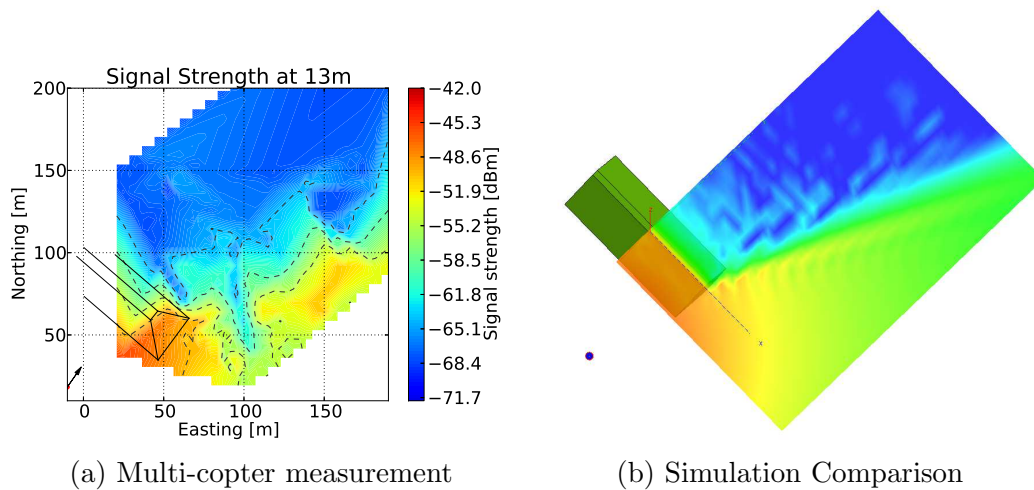


Figure 4.23: Diffraction flight path simulations at 50m and 200m.

Figure 4.24: Horizontal plane measurement made by LS of SA compared to the FEKO[®] model. Measurements directly behind the berm are mostly in the noise floor. However, the propagation around the berm is still informative.

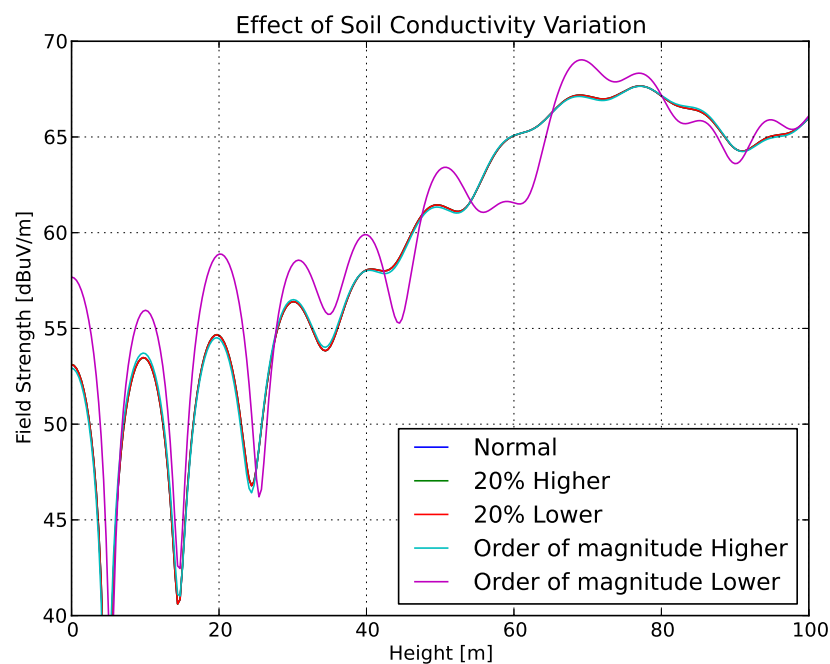


Figure 4.25: Berm loss-tangent parameter study. The loss-tangent was varied initially by only 20% up and down. Due to the insignificant change, the loss-tangent was further changed by an order of magnitude. This shows how diffraction over the berm converges with an increase in conductivity.

change the parameters of the soil. Therefore, using the simulation basis developed in this chapter, a parameter study could be done. Initially, the loss tangent was increased and decreased by 20% (see Fig. 4.25). Other than a small increase in the decreasing conductivity trace, the difference was barely visible. Thereafter, the loss tangent was varied by an order of magnitude in each direction. In this case, the increasing conductivity once again did not show any difference. However, the order of magnitude decrease in conductivity showed a significant difference. This indicates a point of convergence with increasing conductivity. Therefore, the variability is a function of the initial parameters. In this case, the parameters were stable for even a 20% change. This means that seasonal variations would have an insignificant effect on measurements and simulations.

4.7 Conclusion

In this chapter, scale measurements were used to verify computational models. At first, an FDTD code was used to represent the scale model accurately. It was found that the problem was electrically too large to be practically simulated. Consequently, infinite approximations were used to simplify the simulation.

As a novel test, a serrated plate was optimised to function as a quasi-infinite ground plane. These infinite approximations made it possible to use a MoM code efficiently. However, these conditions did not allow for accurate modelling of the feeds which caused deviations in comparisons. Eliminating the PEC half-space using a mirror principle made it possible to use a hybrid RL-GO method in FEKO[®]. This method made full-scale 3D berm simulations possible. Previously, full-scale simulations were only possible using infinite 2D approximations. As a result, berm shielding could be investigated where we found that shielding would always be positive as long as both the transmitter and receiver were below the apex of the diffractor. Berm shielding performance was studied by making use of an average shielding metric coupled with our 3D model. We found that the berm only becomes effective (10 dB average shielding) when its height reaches around 10 wavelengths. The minimum shielding was then maintained at around 15 dB at a berm height of 40 wavelengths.

Using the 3D model, planned measurements could be simulated before campaigns. Also, through a parameter study it was found that variation in the dielectric parameters of the soil did not have a significant effect on its simulated diffraction. A similar result was obtained by N. S. Tezel using an infinite 2D berm approximation. With this simulation tool, it is now possible to investigate alternative terrestrial structures. The next chapter will focus on the construction of the aerial vehicle to test these models on a larger scale.

Chapter 5

Multi-copter Metrology Development

5.1 Introduction

Real world propagation measurements are a time-consuming exercise for even a small set of data points. These types of measurements are, usually, done at ground level, shifting the measurement equipment after each measurement. In some cases, a vertical mast is used to increase the height of the measurement antenna. In other cases, ground vehicles can be used to move the antenna if the terrain is suitable. However, due to the limited time and nature of the measurements needed for this campaign it was decided to use an aerial vehicle. Such a vehicle has a much higher height limit, is not affected by the terrain and can be easily automated. The drawback of such a vehicle is mostly its size (small ground plane) and self-generated RFI. In this chapter, the design of a Multi-copter will be discussed. This Multi-copter was used during the Karoo-measurement campaigns to collect the data that is analysed in the next chapter.

The focus of this vehicle will be to gather data for local-propagation and shielding studies. This means that there will always be a transmitting antenna placed at a position of interest. With this known radiator, propagation and shielding information about the terrain and structure under test can be collected. This chapter will consist of multiple parts discussing the key areas of the Multi-copter. The first part is the mechanical structure followed by the electronics used for its flight systems. After that, the receiver and antenna components will be covered including a brief insight into the vehicle's electromagnetic signature and electromagnetic compatibility (EMC) problems. Finally, de-embedding of measurements used for the data in the next chapter will be discussed.



Figure 5.1: Multi-copter CAD model compared to a photo of its realised counterpart. Plastic parts were 3D-printed while metallic structure parts were obtained from a local hardware store.

Part	Frame	Motors	ESC's ¹	Autopilot	Receiver	Battery	Total
Weight	900 g	300 g	100 g	50 g	150 g	320 g	1820 g

Table 5.1: A weight estimate of the Multi-copter with all of its systems on-board.

5.2 Vehicle design

The mechanical structure of the Multi-copter was entirely constructed in-house using a 3D printer and local-hardware supplies. This allowed for a low-cost vehicle while maintaining the similarity to its CEM model. Fig. 5.1 shows a photo of the frame next to its CAD-model. The frame was designed with the antennas integrated into its landing gear. All metallic sections such as the arms of the Multi-copter and the copper-shield discussed further on were all galvanically bonded to the same DC potential.

An open-source flight controller (Ardupilot Mega 2.6 from 3D Robotics [74]) was used as a control system for the Multi-copter. This commercial off-the-shelf (COTS) hardware already has the capability of autonomous flight and therefore expedited development. Also, because of its open-source nature, access to its internal parameters was possible. Fig. 5.2 shows the flight electronics before installation. The flight electronics consists of four main parts: a remote control (RC) unit which receives user input and relays it to the flight controller; a flight controller, that depending on the input of the user, uses its on-board sensors to control the motors; and speed controllers (ESC's) that drive the brushless-dc motors, and finally the motors themselves. All of this is powered by a 3-cell 4200 mAh battery. These are all available as RC-hobbyist equipment, which lowered cost and design time.

The Multi-copter was designed as a 4-rotor vehicle, keeping the wiring simple while maintaining a low-cost to payload-weight ratio. To be able to make a suitable choice for the motors, a weight estimate was made (see Table 5.1).

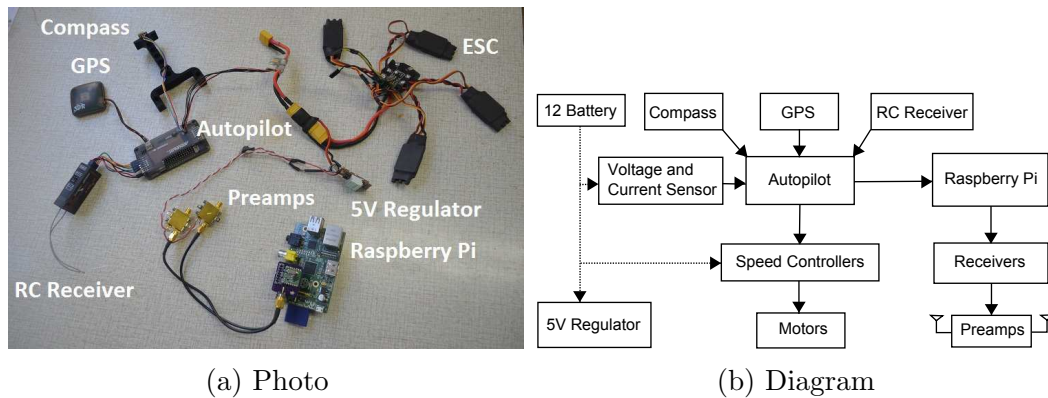


Figure 5.2: Photo and diagram of flight electronics. In (b), the connections between the different electronics can be seen.

Throttle	Current [A]	Watt [W]	Thrust [g]	RPM	Eff. [g/W]
50%	3	35	350	4900	10
65%	4.4	50	420	5400	8.4
75%	5.7	64	530	5900	8.28
85%	7.3	82	630	6500	7.68
100%	8.9	98	720	7000	7.35

Table 5.2: Brushless-dc motor specifications. This data is valid for an MT2216 900 KV motor at 11.1V on an 11x3.7 inch propeller. The rest of the data can be found from the manufacturer's specification sheet in Appendix E.3.

With this information, it can be assumed that the Multi-copter will be able to hover with a total thrust of 1.82 kg. To generate this, each motor needs to deliver 455 g of thrust. With the 10x4.5² pitch propellers, the Multi-copter should hover at about 68% throttle. The specification for a single motor can be seen in Table 5.2.

This is slightly higher than ideal and can be lowered by using longer 12x4.5 pitch propellers. A lower throttle will deliver greater efficiency and a slightly more stable vehicle. Due to availability, the smaller propellers were used for the duration of this dissertation. To calculate the average flight time, the current at hover was used in conjunction with the mAh rating of the battery. This gives a somewhat optimistic flight time of 13 minutes at a constant current of 18.9 A. Factoring in the current drawn by the receiver this estimation lowers to about 12.5 minutes. As a safety margin, the Multi-copter will be used only up to two-thirds of its battery capacity (8 min). This ensures that the batteries are not damaged and allows enough time to plan a safe landing. At 100% throttle the current to each motor would be 8.9 A, adding up to 35.6 A for

²10 stands for the tip to tip length of the propeller whereas the 4.5 indicates the pitch of the propeller

the whole Multi-copter. The battery is capable of discharging at 30 times its capacity, which is 126 A. This means that both the 30 A speed controllers and LiPo battery will be far within their specifications at maximum throttle. At this point, a Multi-copter has been designed which is capable of carrying the intended payload. The next section will cover the receiver in detail.

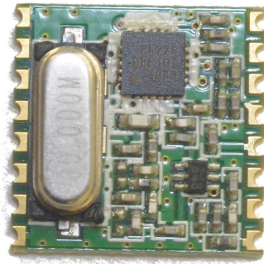
5.3 Receiver development

The Multi-copter was built as a metrological device with dual-embedded antennas feeding into two receiver systems. Because of the obvious weight restrictions a light design was preferred. For propagation studies, a known signal is transmitted at a known position and then received at a second known position. The propagation characteristics are then inferred from the change in signal between the transmitter and receiver. The transmitted signal consists of a single-frequency carrier, sweeping-frequency carrier or a time-domain pulse. A single carrier frequency simplifies the receiver that is needed in the Multi-copter as well as post-processing of data. However, this limits the investigation to that single frequency for the entire flight. It is possible to sweep the fixed carrier during the flight. This will, however, reduce the spatial resolution of each frequency and could introduce measurement inaccuracies if the transmitter and receiver are not properly synchronised. Nonetheless, it is possible to achieve continuous frequency data while preserving the spatial resolution by using a pulse generator and a transient real-time analyser as the receiver on the vehicle. However, this dramatically increases the complexity of the on-board receiver.

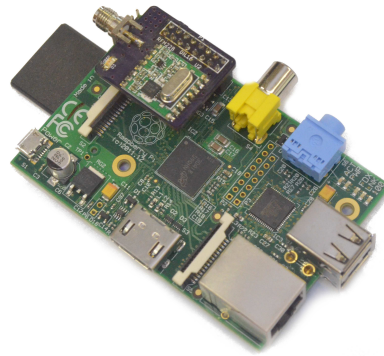
For most of our propagation measurements, the single-carrier frequency scheme was used. This preserved simplicity and eliminated extra unknowns when de-embedding the vehicle from the measurements. Sec. 5.3.1 discusses the narrowband receiver used for our propagation studies. Additionally, a miniaturised lightweight real-time transient analyser is also investigated and developed in Sec. 5.3.2.

5.3.1 Narrowband receiver

The nature of narrowband propagation measurements involves a fixed carrier being transmitted at a single frequency. Therefore, a sensor that can receive this carrier and measure its signal strength would be sufficient. It was decided to use a transceiver module, RFM22B seen in Fig. 5.3a, which incorporates a digital receive-signal-strength-indicator (RSSI) with a 0 to -120 dBm range. Along with an 8-bit analog-to-digital converter (ADC), the module can deliver up to 0.5 dB resolution with a linear trend. The primary interface to this module is through a serial peripheral interface (SPI). To keep the interface to the flight controller as well as the receiver simple it was decided to use a



(a) RFM22B transceiver module



(b) Raspberry-Pi SBC

Figure 5.3: Photos of the RFM22B transceiver module. In (b), the transceiver is shown on its daughter board attached to the Raspberry Pi.

low-cost single-board computer (SBC) called Raspberry Pi. This computer has on-board digital input and output pins which support SPI. Fig. 5.3b shows the SBC with the receiver connected on to its digital output pins. The connector board between the SBC and receiver was designed in-house and can be found in Appendix E.1.

One of the main advantages of this vehicle is the speed at which it can collect data. This is because the SBC continuously interrogates the availability of new data using multiple threads. As soon as new data is available it is stored on an SD card, connected to the SBC. The software uses individual threads for the flight controller, receiver and data storage. A diagram illustrating the operation of the software can be seen in Fig. 5.4. The software starts by creating the two main threads, a receiver thread and flight controller thread. The receiver thread and flight controller thread continuously poll their devices for new data. As soon as new data becomes available, flags are set. The storage thread, also the main thread, will keep a record of the latest available data and monitor the flags. As soon as both the receiver and flight controller indicate the availability of new data, the storage thread will store the data on the SD card. The multi-threading makes it possible for the receiver thread to average the signal data while other data is being collected. However, care was needed to prevent race conditions. Therefore mutexes³ were used throughout the data accessing portions of the code. The script that was used for the data collection during measurements can be found in Appendix E.4

To ensure accurate measurements, it was necessary to calibrate the receiver. This was done using a signal generator, checked against a calibrated spectrum analyser, for each of the calibration frequencies. With the power level known

³Mutual-exclusion mechanism ensuring that no two concurrent processes access a critical section at the same time. The critical section here represents a section of code modifying or using variables.

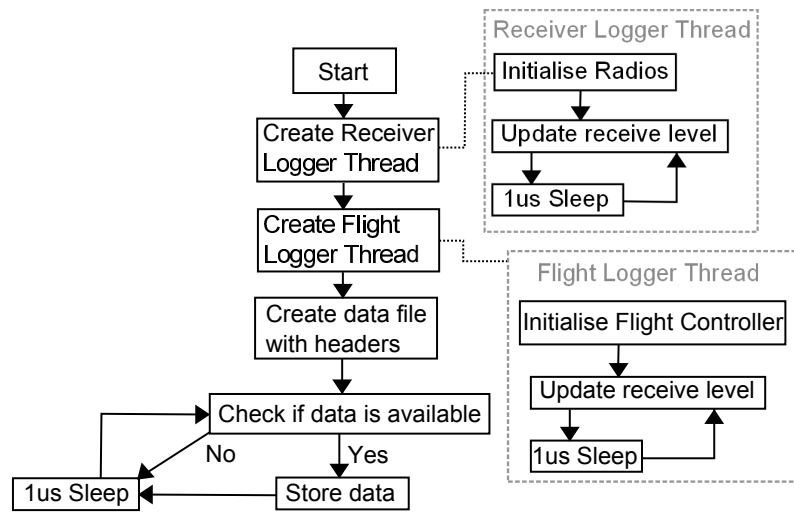


Figure 5.4: Software diagram of the logging script running on the Raspberry-Pi. This python script is initialised before each flight and is only terminated after the flight has ended.

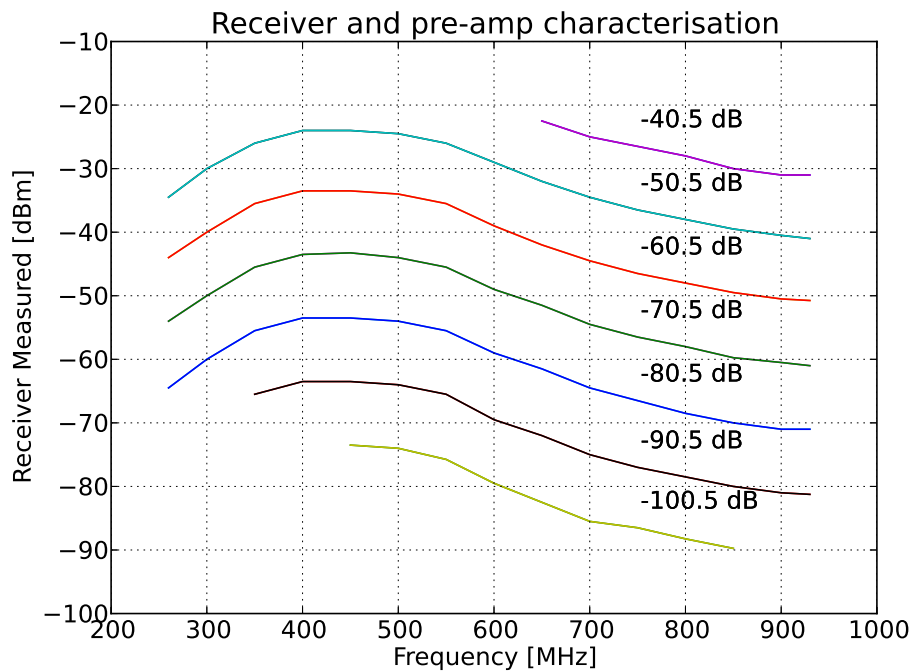


Figure 5.5: The calibration file used during post-processing for a pre-amp receiver pair. This file was created using a known signal-generator as an input while logging the output of the receiver.

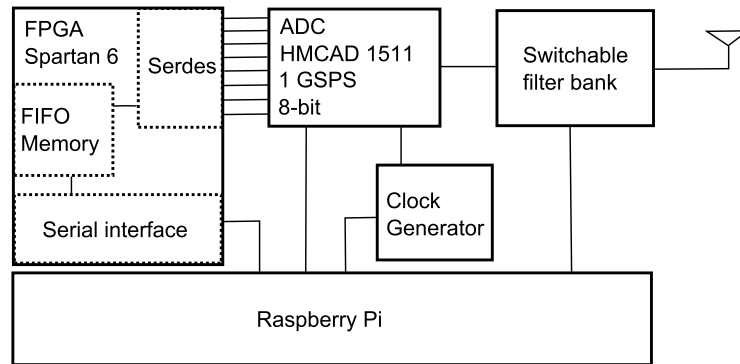


Figure 5.6: System diagram of the real-time transient analyser.

to the point of the receiver connector, it was possible to generate a calibration file. Such a file contains the power level transmitted into the receiver versus the power level measured with the receiver. An example of such a calibration file can be seen plotted in Fig. 5.5. This data, using interpolation, is then used during post-processing to remove the effect of the receiver.

5.3.2 Wideband receiver

A real-time transient analyser has the capability of sampling short duration pulses. With the frequency content of the pulse at transmission known, it is possible to characterise a broad range of frequencies by comparing it to the spectral content of the received pulse. Relative measurements are also possible where the pulse is not exactly characterised. An example of such a measurement is sparking on power lines discussed in Kibet-Langat [6] and Groch [5] where the pattern of the broadband radiation pattern can be characterised. Such wideband emissions are cumbersome to measure using a narrowband receiver.

In this investigation, a compact and lightweight direct sampling transient analyser is designed and tested. Such a receiver uses an analogue front-end to filter a specific Nyquist zone that is sampled by a high-speed sampling ADC. Data from the high-speed ADC is buffered using high-speed memory. This memory is in most cases limited and needs to be cleared before measuring the next pulse. A simplified diagram of the transient analyser can be seen in Fig 5.6.

It was decided to develop the transient analyser for measuring power-line sparking. An in-house 12 kV line with a spark-gap was used to generate a power-line sparking scenario. The sparking was then recorded using a full-size transient analyser developed in house called RaTTy [13]. Measurements revealed that a bandwidth of 500 MHz would be sufficient and that the emissions were powerful enough to be received without any amplification stages, see Appendix F.1. With this bandwidth, The HMCAD1511 1 GSPS 8-bit ADC with a full power bandwidth of 650 MHz from Hittite was chosen. To

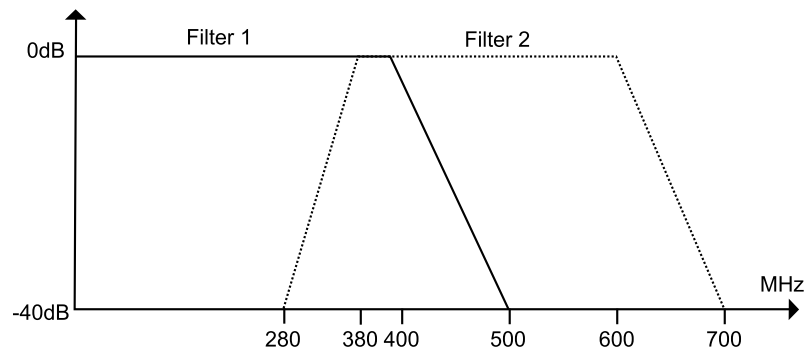


Figure 5.7: Front-end switchable filter-bank configuration.

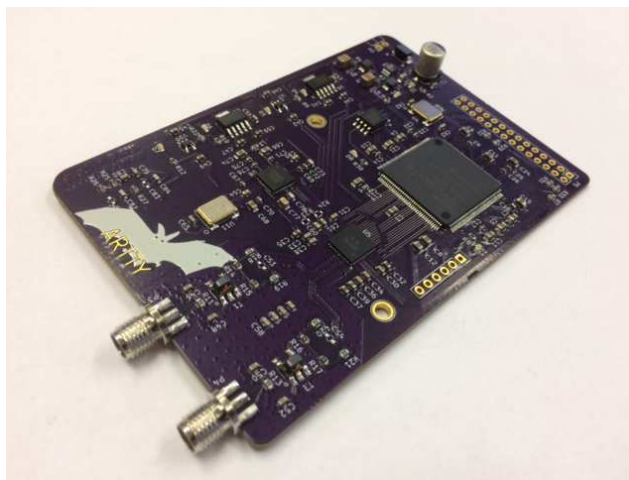


Figure 5.8: Photo of the digital board containing the FPGA, ADC and clock generator.

cover the full power bandwidth, the front-end would consist of two switch-able filters with specifications illustrated in Fig. 5.7. The ADC will feed its data through 8 differential LVDS lines at 1 Gb/s into a Xilinx Spartan 6 FPGA. Each data channel will be de-serialised and then stored into a FIFO buffer. A state-machine on-board the FPGA will make the data available through a serial interface that will then be downloaded onto the Raspberry-Pi for storage. Schematics, PCB layouts and VHDL code can be found in Appendix F.2. The manufactured digital ADC/FPGA board can be seen in Fig. 5.8. An implementation of the switch-able filter bank can be seen in Fig. 5.9.

Similar to the narrowband receiver, calibration will be necessary to remove errors introduced between the ADC and front-end input. However, due to unforeseen manufacturing errors signal integrity policies on the main-digital board were severely compromised. As a result, the system could never be fully implemented on the full Multi-copter platform. Nevertheless, sufficient data was captured using the narrow band receiver in previous campaigns to enable off-site propagation studies. Test measurements on the compromised system



Figure 5.9: Photo of the unshielded switchable filter bank. Filters are implemented using lumped elements.

can be seen in Appendix F.2.

5.4 Antenna integration

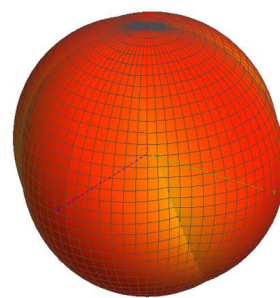
5.4.1 Antenna configuration

To feed the measured power into the receiver, discussed in the previous section, an antenna is needed. Because Multi-copters are small compared to the measured wavelength, antenna design becomes an important part of the system. A Multi-copter tends to use its entire platform to control its position. As a result, an antenna attached to the vehicle will need to either compensate for its movement or the logged data needs to be post-processed for each antenna position. The first option can be achieved by using an antenna gimbal. However, changing the orientation of the antenna about the vehicle will change the antenna pattern. Given enough time it might have been possible to correct for each gimbal position. However, in the end it was decided to go for a much simpler static-antenna design. The problem however with a static antenna is its variability in certain orientations. This becomes a problem when the source is located through the null of the Multi-copter antenna. To counter this, a dual antenna design was chosen. Each antenna is placed at a 45-degree angle, 90-degrees rotated around the vehicle as seen in Fig. 5.10a.

Each antenna has its own receiver and will be continually logged. The antenna used in the final measurement is only selected in post-processing. This is based on the antenna pattern, vehicle orientation and relative direction to the source. Fig. 5.10b shows how the two antennas, measured individually, forms an almost isotropic antenna pattern. An added advantage of the static configuration is that the antennas themselves can be embedded into the structure of the Multi-copter. Using 3D printing, the legs of the Multi-copter could be designed within a CEM package and directly exported and printed in the laboratory (See Fig 5.10a). This meant that the constructed antenna leg should behave very much like the simulated model.



(a) Antenna Layout



(b) Antenna Pattern

Figure 5.10: Photo showing the layout of the dual antenna configuration. In (b), the isotropic antenna pattern formed by an idealised dual-configuration can be seen.



Figure 5.11: A photo in-flight showing the grounded-copper shield enclosure. The compass and GPS due to the nature of their operation are still placed on the top of the cube. Photo was taken by Dr. P. G. Wiid.

The antenna pattern is a function of the wires and metallic structures on the vehicle. The complexity of these structures makes them almost impossible to simulate in a CEM package. Added to this, after each battery change, the position of the wires might shift slightly which will in turn change the antenna pattern. Therefore, to maintain a stable antenna pattern throughout a measurement campaign, it became necessary to have a stable metallic environment. Consequently, it was decided to enclose most of the on-board wiring into a grounded copper-cube seen in Fig 5.11. This meant that perturbations to cabling inside the vehicle could be mostly isolated from the antennas. The global positioning system (GPS) and compass subsystems were still placed on the top of the shielding enclosure to ensure their operation.

5.4.2 Broadband antenna design

A drawback of the integrated antennas is that they are not easily exchanged. With limited time on campaigns, it was important to design a single antenna that would cover a reasonable amount of bandwidth. The bandwidth of the receiver is 260 MHz to 960 MHz. This is not achieved easily on a platform the size of a Multi-copter. However, with some compromise, a design was attainable using a resistively-loaded monopole. This design works for the mentioned

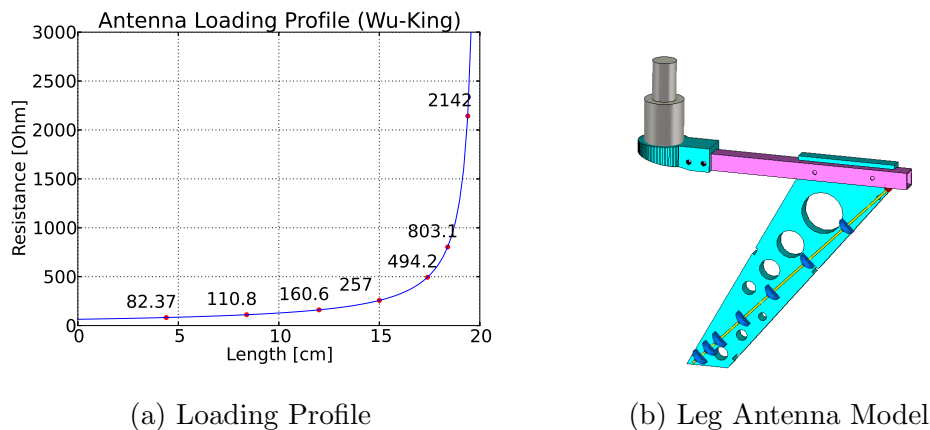


Figure 5.12: The loading profile of the monopole can be seen in (a). In (b) a picture of the initial simulation model, used in CST MWS[®], can be seen. One half of the leg here is removed to reveal the antenna with its loaded elements.

bandwidth with limited efficiency. In the interest of saving weight and maintaining a rigid landing structure for the Multi-copter, the integrated antennas were limited to a length of 20 cm. Using this length the monopole is discretely loaded with resistive elements, using a Wu-King [75][76][77] profile seen in Fig. 5.12a. To ensure the proper functioning of the design, a CEM model was constructed which can be seen in Fig. 5.12b. In this image, it is possible to illustrate how the Multi-copter arm was used as a ground plane. The antenna here is encapsulated in a PLA plastic that also functions as the Multi-copter landing gear. Encasing the element inside a dielectric would also help lower the minimum frequency of the antenna. After investigating the simulated model, it became apparent that the loaded antenna would be well matched to a 200 Ω system. A 4:1 impedance transformer was used to match the current 50 Ω system to the 200 Ω antenna. It was not possible to replicate the simulated performance. The characteristics of the transformer decreased the overall measured performance of the antenna. Because of this it was decided to use the antenna in its unmatched form. To compensate for this mismatch, as well as the inefficiency of the loaded monopole, a broadband low noise amplifier (LNA) (ZFL-1000LN+ see Appendix E.3) was added between the antenna and receiver. This simplified simulation expedited the optimisation and testing of the antenna to the point where the full vehicle could be simulated.

After confirming the operation of the antenna, it was simulated with the entire vehicle (see Fig. 5.13a). This included a copper shield enclosing the electronics. The simulated input reflection compared to its measurement can be seen in Fig. 5.13b. The agreement between the two datasets was deemed good for this complex structure. Only a slight deviation is noticed, which is less than 1 dB. This was attributed to non-modelled details as well as the loading effect of the environment during measurement. With confidence in

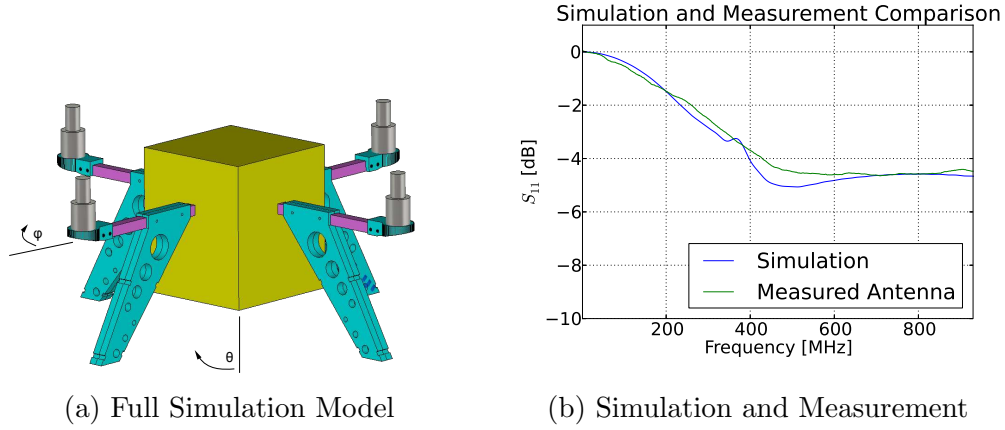


Figure 5.13: Image of the full Multi-copter CEM model. Here the simulated input reflection of the antenna is compared to its measured equivalent.

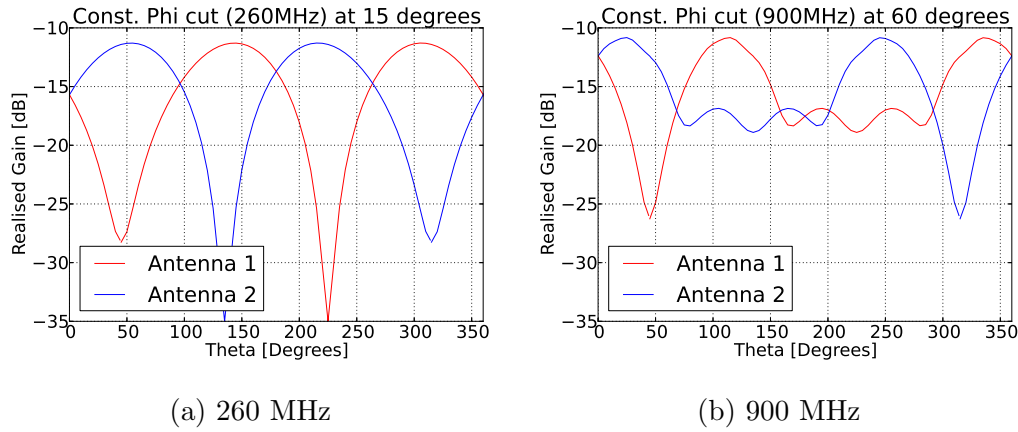


Figure 5.14: Realised gain of Multi-copter antennas on a constant phi cut. Cut locations were chosen to align with the deepest nulls of each pattern. During measurement extraction, the data from the antenna with the highest realised gain will be used. Here the difference with increasing frequency can be seen.

the CEM model of the antennas on the Multi-copter, it is now possible to simulate its spherical realised-gain patterns. For the measurements during the campaign, discussed in the next chapter, these patterns (see Appendix E.2) will be used to de-embed the data. These patterns become more dynamic at the higher frequency bands (900MHz). This causes the sizes of the nulls to increase leading to a less isotropic antenna pattern. This can be seen illustrated in Fig. 5.14. The superposition of the two antennas illustrates its benefit. The null point of each antenna is covered by the peak of the other. During post-processing, the data from the antenna with the best realised-gain will be used.

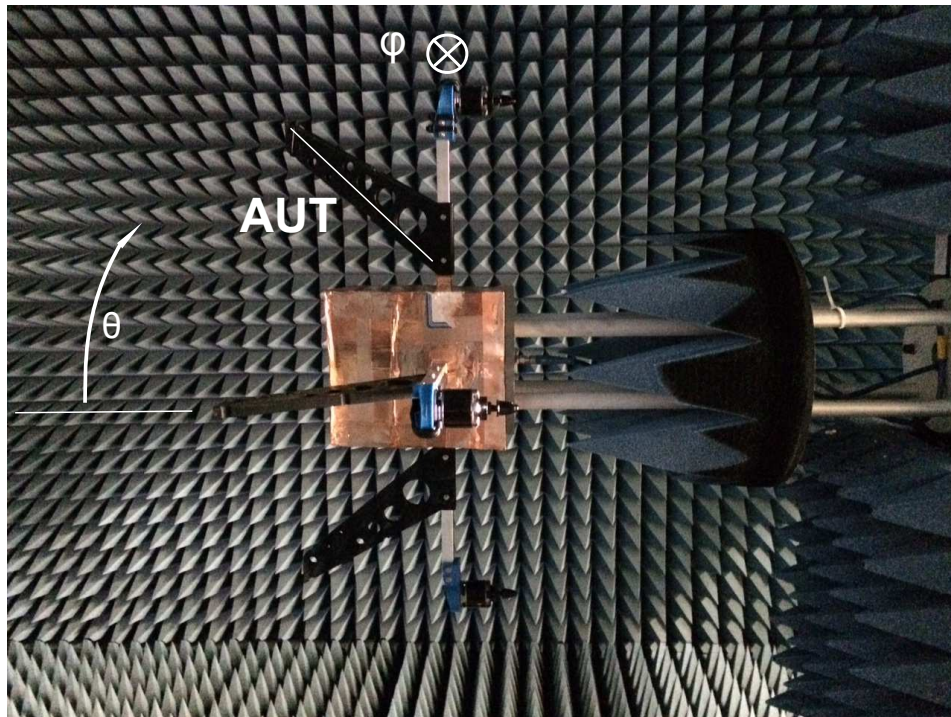


Figure 5.15: Multi-copter mounted on the near-field scanning device. The highlighted (white line) is the active antenna during the measurement.

5.5 Antenna verification

As we saw in the previous section, the dual antenna configuration on the vehicle was designed. This section shifts its focus to the verification of the realised antenna performance. In order to achieve this, far-field patterns need to be measured. Verifying the antenna pattern is a key goal in for the Multi-copter design.

The complex arrangement of subsystems and wiring was specifically enclosed to ensure the antenna patterns were simple enough for CEM simulation tools. Measurements will be used to verify the CEM model of the Multi-copter. The computational model will then be used for further investigations and the de-embedding of actual measurements.

An in-house anechoic chamber with a near-field scanner was used for the measurements. The minimum frequency of the chamber gave us the ability to verify the Multi-copter radiation pattern from 700 MHz upwards. The vehicle will practically only be used up to 960 MHz. This section will therefore focus on comparisons at 700 MHz and 900 MHz. The vehicle can be seen mounted on the spherical near-field scanner (Fig. 5.15) with its top facing to the right. Plastic bolts through the top of the frame are used to secure the vehicle in place. The antenna cable can be seen routed through a grounded bulkhead connector on the top of the vehicle. The mounting and cable interface location

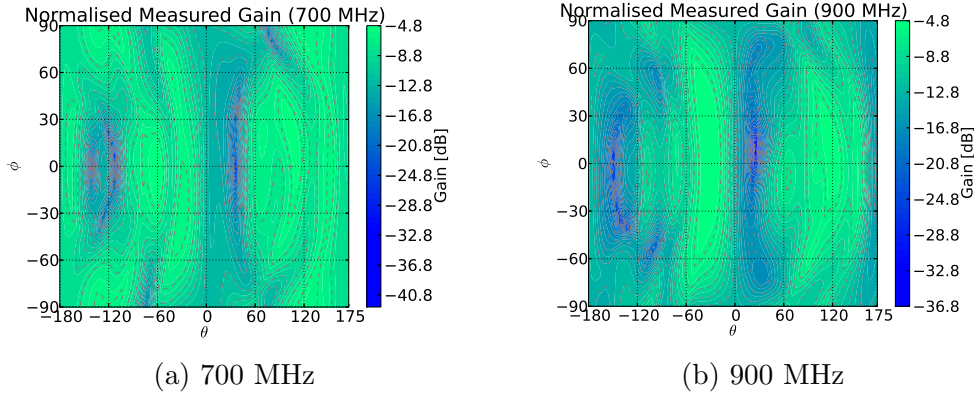


Figure 5.16: Measured 2D theta/phi antenna patterns. Data transformed to far-field from a near-field scanning measurement.

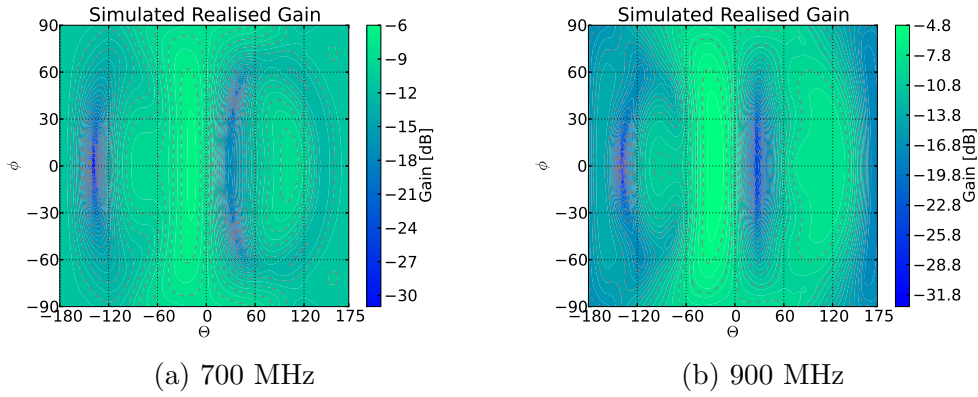


Figure 5.17: Simulated 2D antenna patterns using CST MWS[®].

were specifically chosen in order to minimise the effect on the measured antenna pattern.

For each measurement, the full 3D near-field is scanned using a theta-phi coordinate system [78]. The near-field is then transformed into the resulting far-field pattern. The resulting measurements of a single antenna pattern can be seen in Fig 5.16a and 5.16b. The measurement can be compared to their respective CEM simulations in Fig 5.17a and 5.17b. A distorted monopole pattern is visible. The distortion is a function of the electrically small and unconventional ground-plane. The simulated major broadside-beams compare well with their measured equivalents. There are differences visible only in the local null areas where the measurements reveal fine features not present in the simulations. These features are most probably a result of manufacturing tolerances. A detailed investigation on this subject is not warranted as this structure will not affect our measurements due to the antenna selection scheme during post-processing.

It is also worth mentioning that at θ angles of $\pm 180^\circ$ some differences can

be seen. These angles relate to a scenario where the structure of the near-field scanner is now between the antenna-under-test (AUT) and probe. This is a well-known problem with near-field scanning. To keep the resulting error to a minimum, antennas are mounted to produce the lowest emissions along this path. The transformation from near to far-field could cause these errors to propagate over the rest of the angles. To test the severity of the errors, the data was truncated around the affected near-field angles while observing the effect on the transformed far-field values. Truncation on these angles did not cause significant changes. It is also possible to include relevant parts of the spherical near-field scanner in the simulation. However, the simulation would be computationally too expensive at a frequency of 700 MHz.

It is now possible to use the verified model for further design and investigation into the antenna patterns. This was made possible by enclosing all of the subsystems into a shielded enclosure to simplify the CEM model. The shielded enclosure also resulted in temporally stable antenna performance making the de-embedding of measurements more reliable over subsequent campaigns.

5.6 Electromagnetic signature and EMC recommendations

As mentioned earlier, the second problem with a metrology vehicle is its self-generated RFI. This could increase the minimum noise level of a sensitive measurement. Also, it is possible for the electronics to generate its own carriers, which could be mistaken for the target carrier. This dissertation made no attempt to decrease the self-generated RFI but did, however, seek to identify it. In addition to measuring the RFI signature of the vehicle, some recommendations for further development regarding RFI are given. The primary sources of on-board noise are the motor drivers and microprocessor clock. It can be recalled that the current in the motor driving wires varies between 18 A and 30 A. Because of this significant amount of current, care was taken to twist these cable pairs and route them close to the grounded metallic-structure where possible in order to reduce coupling. In the case of the 3-phase motor wires, the cabling was routed through the inside of the Multi-copter arms. These currents could be reduced by using a higher voltage battery to drive the motors. There are at least four micro-controller devices in the system. The flight controller which runs at 16 MHz, the receiver-board clock at 1 MHz, a GPS with a 10 MHz clock and finally the Raspberry-Pi SBC with a 700 MHz clock. It should be noted, even though these sources were not eliminated, a spectrum check was done before choosing the measurement frequencies in the following campaign. Additionally, raw measurements were checked during post-processing for signs of RFI interference.

As a means to quantify the self-generated RFI, a reverberation chamber

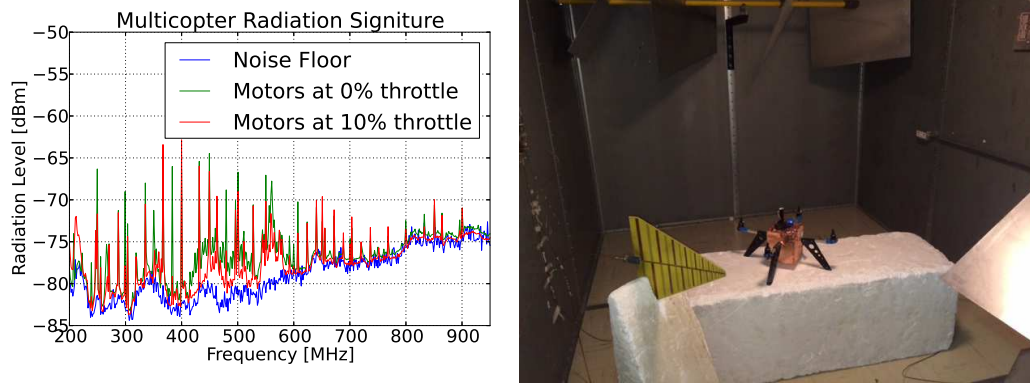


Figure 5.18: Measurement of Multi-copter RFI signature using a reverberation chamber. The measurement can be seen in (a) with a photo of the Multi-copter in the chamber in (b).

measurement was conducted in the Electronic Engineering department's high-frequency laboratory. In this measurement the Multi-copter was placed in a shielded room with stirring paddles, known as a reverberation chamber. An LPDA was used as the receiver to a spectrum analyser. The frequency range, 200 MHz to 950 MHz was then scanned multiple times at 72 paddle positions (with thanks to fellow Post Doctoral student J. A. Andriambeloson). After applying the cable losses, antenna efficiency and chamber loss to the measurements, the power radiated by the Multi-copter could be plotted (see Fig. 5.18). In this experiment three different measurements were made. Firstly the background was measured to obtain a baseline. After this, the Multi-copter was measured at 0% throttle, motors at a standstill. Finally, the Multi-copter was measured with the motors at 10% throttle. At 0% throttle, eight frequency scans were possible, due to the lower current consumption. At the higher throttle value, the number of frequency scans was limited to conserve battery throughout the measurement. The effect of this can be seen in the slightly less measured RFI. The analysis indicates a large number of RFI peaks, some up to about -63 dBm. When the motors are activated, an increasing amount of RFI can be seen below 250 MHz. Campaign measurements were all done at frequencies higher than this. Therefore, noise as a result of the motors at differing throttle levels are not of too much concern.

To help reduce the RFI being radiated from the vehicle, a shielded enclosure approach is proposed. This would mean that all of the apertures, leading from the shielded cube to the outside, be closed off. Therefore, all wires for the motors, GPS and compass would have to be routed through feed-through capacitors. Additionally, the battery-compartment lid would need to be designed to seal after each flight. Having a shielded enclosure will help reduced energy coupling onto the antenna from on-board sources, except the GPS, which would still be mounted outside. However, it is still possible for energy to couple from the wires and surfaces on the inside of the enclosure onto

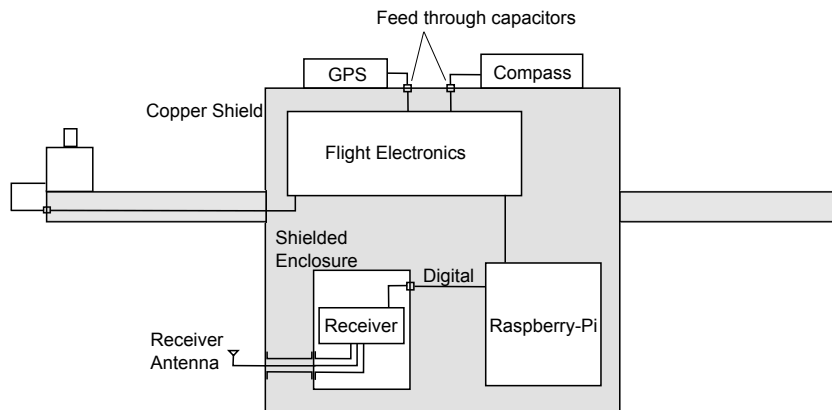


Figure 5.19: A simplified schematic of the Multi-copter illustrating the clean-room dirty-room concept. The receiver is shielded from the internally-generated RFI, illustrated in grey.

the receiver. Here it is proposed to build a shielded box for the receiver itself where the data lines enter through filtered mounts. This will form a "clean" room inside of a "dirty room", see Fig. 5.19. However, for the purpose of this study RFI did not affect measurements. Therefore, these modifications were not pursued further as it did not have significant bearing on the results.

5.7 Measurement extraction

5.7.1 De-embedding received power from Multi-copter

As mentioned earlier, the propagation and shielding measurements involve a fixed transmitter with the Multi-copter as the receiver. The nature of the flight path will cause the orientation relative to the transmitter to change. Additionally, the antenna, pre-amp and receiver calibration are dependent on the frequency of the measurement. These are all factors that need to be taken into account when extracting the measurement.

Starting with the realised-gain antenna pattern, the spherical coordinate of each antenna to the transmitter needs to be calculated. This is possible due to the GPS, compass, barometer and attitude sensors on the Multi-copter. To begin with, the coordinates of the Multi-copter and the transmitting antenna are converted to Universal Transverse Mercator (UTM), easting and northing, format. With the positional information now in the Cartesian domain, it is possible to calculate the position of the source relative to the vehicle using Eq. 5.7.1.

$$\mathbf{X}_r = \mathbf{X}_{tx} - \mathbf{X}_v \quad (5.7.1)$$

where

- \mathbf{X}_r is the relative position vector of the source to the vehicle
- \mathbf{X}_{tx} is the position vector of the source
- \mathbf{X}_v is the position vector of the vehicle

Using rotational matrices [79], see Eq. 5.7.2 and Eq. 5.7.3, the transmitting point is rotated according to the roll, pitch and yaw information of the Multi-copter. During this rotation, the vehicle itself is kept static in orientation and position. West is taken as the positive x direction while North is taken as the positive y-axis

$$\mathbf{R}_{tot} = \mathbf{R}_z \mathbf{R}_x \mathbf{R}_y \quad (5.7.2)$$

where

- $\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(-\alpha) & -\sin(-\alpha) \\ 0 & \sin(-\alpha) & \cos(-\alpha) \end{bmatrix}$
- α represents pitch where nose-up translates to a positive value
- $\mathbf{R}_y = \begin{bmatrix} \cos(-\beta) & 0 & \sin(-\beta) \\ 0 & 1 & 0 \\ -\sin(-\beta) & 0 & \cos(-\beta) \end{bmatrix}$
- β represents roll where banking right translates to a positive value
- $\mathbf{R}_z = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- γ represents heading (yaw), turning clockwise is positive

$$\mathbf{X}_{rot} = \mathbf{R}_{tot} \mathbf{X}_r \quad (5.7.3)$$

where

- \mathbf{X}_{rot} represents the relative position of the source given a static vehicle orientation and position.

The final coordinates phi and theta of the source point can now be calculated using Eq. 5.7.4⁴ and Eq. 5.7.5 respectively. It should be noted that each antenna has a 45-degree offset from the 0-degree heading. This is easily factored in for each antenna as seen in Eq. 5.7.6.

⁴The arctan used in this formula handles the calculation according to the appropriate quadrant

$$\phi = -\arctan \frac{\mathbf{X}_{\text{rot}}^y}{\mathbf{X}_{\text{rot}}^x} \quad (5.7.4)$$

where

- ϕ is the horizontal spherical angle, clockwise from the front of the vehicle.

$$\theta = -\arcsin \frac{\mathbf{X}_{\text{rot}}^z}{|\mathbf{X}_{\text{rot}}|} \quad (5.7.5)$$

where

- θ is the vertical spherical angle, positive downwards from the horizontal.

With ϕ and θ it is possible to calculate the antenna pattern relevant to the data point.

$$\begin{aligned} \phi_{ant1} &= \phi - \frac{\pi}{4} \\ \phi_{ant2} &= \phi + \frac{\pi}{4} \end{aligned} \quad (5.7.6)$$

With the realised gain, G_{real} , extracted for each antenna, the pre-amp and receiver needs to be de-embedded. For simplicity, the receiver was calibrated along with the pre-amp. Using a graph similar to that seen in Fig. 5.5 G_{rec} is obtained. Polarisation loss also needs to be accounted for, especially because of the 45-degree antenna configuration. Polarisation loss is a dynamic function of both the transmitting antenna orientation as well as the Multi-copter yaw, pitch and roll. It is possible to assume that there will be a constant polarisation loss of 3 dB for small roll and pitch angles. However using the given transmitter and two on-board antennas orientation vectors, the theoretical polarisation loss factor (PLF) can be calculated as in Eq 5.7.7.

$$PLF = |\boldsymbol{\rho}_s \cdot \boldsymbol{\rho}_r|^2 \quad (5.7.7)$$

where

- $\boldsymbol{\rho}_s$ is the vector representing the transmitter polarisation vector
- $\boldsymbol{\rho}_r$ is the vector representing the receiver polarisation vector

The polarisation vectors are defined by Eq. 5.7.8.

$$\boldsymbol{\rho} = a\mathbf{x} + b\mathbf{y} + c\mathbf{z} \quad (5.7.8)$$

where

- $a\mathbf{x}$ represents the East to West vector where due West is positive

- by represents the South to North vector where due North is positive
- $c\mathbf{z}$ represents the vertical vector where down is positive

The source polarisation vector, ρ_s , is static and can easily be defined using the latter equation. However, the receive polarisation vectors are dynamic with respect to the yaw, pitch and roll of the Multi-copter. To find ρ_r the initial vector for each antenna is defined in Eq. 5.7.9 and 5.7.10 as ρ_{r1i} and ρ_{r2i} respectively.

$$\rho_{r1i} = \frac{\mathbf{x} + \mathbf{y} + \mathbf{z}}{\sqrt{3}} \quad (5.7.9)$$

$$\rho_{r2i} = \frac{-\mathbf{x} + \mathbf{y} + \mathbf{z}}{\sqrt{3}} \quad (5.7.10)$$

These vectors were defined for the Multi-copter in its initial orientation where the yaw, pitch and roll are zero. To transform the receiving vectors to ρ_{r1} and ρ_{r2} , ρ_{r1i} and ρ_{r2i} are multiplied by the rotation matrix defined earlier, see Eq. 5.7.11.

$$\rho_{rn} = \mathbf{R}_{tot} \rho_{rni} \quad (5.7.11)$$

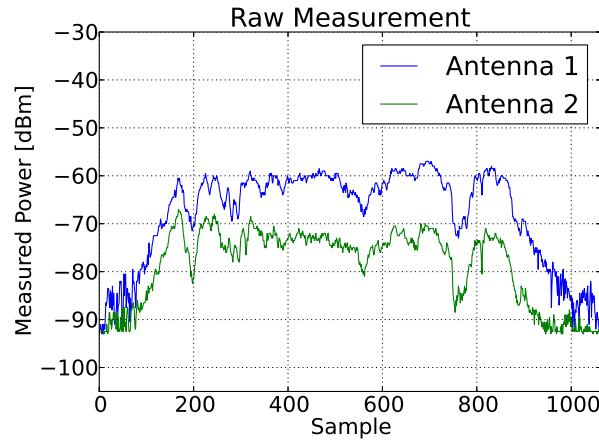
With the rotated polarisation vectors it is possible to find the PLF using Eq. 5.7.7. Finally, the actual power level at the measurement point is calculated using Eq. 5.7.12.

$$P_{actual} = P_{meas} + G_{real}(\phi_{ant}, \theta_{tx}) + G_{rec}(f) + L_{pol} \quad (5.7.12)$$

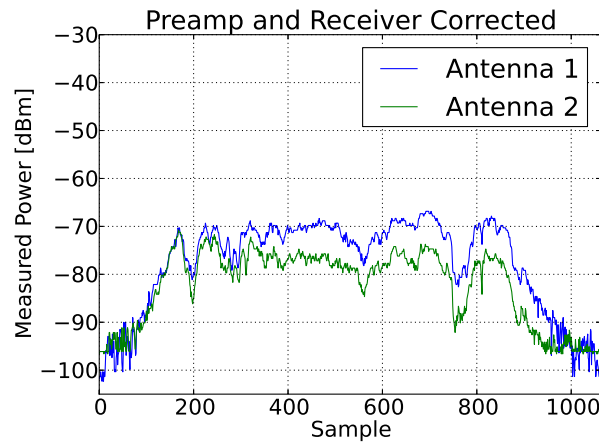
where

- P_{actual} is the actual power at the measured point
- P_{meas} is the power measured at the data point before extraction
- L_{pol} is the polarisation loss factor

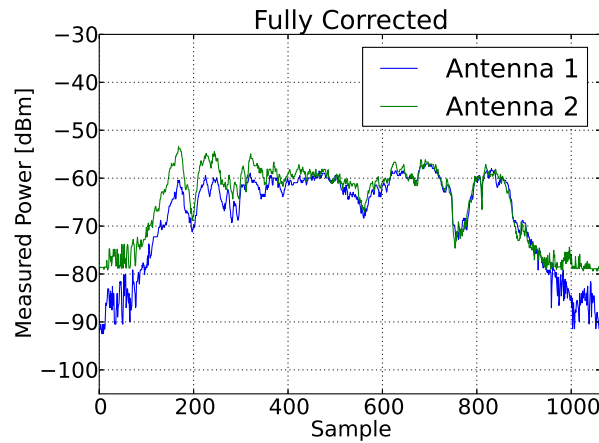
To illustrate the procedures, a practical measurement is investigated. The raw data for each of the two antennas can be seen in Fig. 5.20a. Similarly Fig. 5.20b shows the data after being corrected for the receiver calibration and the pre-amp gain. In the latter comparison, a deviation is visible between the two antenna measurements. This difference is a function of the Multi-copter orientation relative to the source. This is corrected using the routine discussed earlier in this section, see Fig. 5.20c. It should be noted that the antenna pattern was only calculated using an FDTD simulation. This causes some slight errors that can be seen in the start and finish of the measurements. Another explanation of this deviation can be found in the uncertainty of direction for the incoming ray. This was a typical diffraction measurement behind the berm



(a) Raw



(b) Receiver Correction



(c) Fully De-embedded

Figure 5.20: De-embedding measurements using receiver calibration data and simulated antenna patterns. Deviation between the fully extracted data in (c) is a result of slight simulation deviations together with the uncertainty of the principle incoming signal's direction.

which means that the start and end of the measurement were made in its shadowed area. As a result, the direction of the measured incoming signal is not the same as the direct line to the transmitter. It should be noted, the correlation of the two patterns after correction is only used for checking the integrity of the measurement vehicle. Practically, the patterns are used to select the antenna with the best sensitivity during post-processing to extract the final result. This strategy is valuable in the case where there is uncertainty in the direction of the incoming ray due to the quasi-isotropic antenna pattern.

5.7.2 Polarisation estimation of an unknown source

In the previous section the de-embedding of the signal at the Multi-copter was demonstrated. This process required that the polarisation of the transmitter was known. However, there are some applications where it would be beneficial to measure a source where the polarisation is not known. One such application is the characterisation of a single antenna or array of antennas. With the current dual antenna configuration polarisation information is not readily available. Additionally, the receiver system currently does not support relative phase discrimination between the two antennas. The current configuration was designed for RFI shielding and propagation measurements that necessitated the quasi-isotropic antenna pattern while sacrificing polarisation discrimination. However, assuming an accurate de-embedding of the scalar antenna power values it is possible to extract a limited amount of polarisation information. Let us define the power measured at antenna 1 and 2 as S_1 and S_2 respectively. These antennas each have a rotated polarisation vector defined earlier in Eq. 5.7.11 as $\boldsymbol{\rho}_{r1}$ and $\boldsymbol{\rho}_{r2}$. We define the unknown source-antenna polarisation vector as $\boldsymbol{\rho}_s$. Assuming accurate calibration and a successful de-embedding of each antenna from the Multi-copter, Eq. 5.7.13 should hold.

$$S_1 - S_2 = PLF_r \quad (5.7.13)$$

PLF_r is defined in Eq. 5.7.14 as the difference between the polarisation losses of each antenna.

$$PLF_r = |\boldsymbol{\rho}_s \cdot \boldsymbol{\rho}_{r1}|^2 - |\boldsymbol{\rho}_s \cdot \boldsymbol{\rho}_{r2}|^2 \quad (5.7.14)$$

From Eq. 5.7.13 and 5.7.14 and expanding $\boldsymbol{\rho}_s$, $\boldsymbol{\rho}_{r1}$ and $\boldsymbol{\rho}_{r2}$ Eq. 5.7.15 is obtained.

$$S_1 - S_2 = |a_0a_1 + b_0b_1 + c_0c_1|^2 - |a_0a_2 + b_0b_2 + c_0c_2|^2 \quad (5.7.15)$$

where

- a_0, b_0, c_0 are the constituents of $\boldsymbol{\rho}_s$
- a_1, b_1, c_1 are the constituents of $\boldsymbol{\rho}_{r1}$

- a_2, b_2, c_2 are the constituents of $\boldsymbol{\rho}_{r2}$

By solving the relationship between a_0, b_0, c_0 the polarisation vector of the source can be de-embedded. To achieve this, we substitute the known antenna vectors into Eq. 5.7.15. These vectors are selected in line with the coordinate system, while maintaining the relative orientation of the two antenna vectors, in order to simplify the equation for practical purposes.

- $a_1 = 0, b_1 = 1, c_1 = 1$
- $a_2 = -1, b_2 = 0, c_2 = 1$

$$S_1 - S_2 = b_0^2 + 2b_0c_0 - a_2^2 \quad (5.7.16)$$

With this choice of antenna vectors, Eq. 5.7.15 can be seen to simplify into Eq. 5.7.16. To obtain the source polarisation vector, the relationship between the constituents of $\boldsymbol{\rho}_s$ need to be established. With only S_1 and S_2 it is difficult to solve the relationship between a_0, b_0 and c_0 . However, if we can assume the AUT is aligned with the horizontal plane ($c_0 = 0$) it is possible to extract some relationships for a_0 and b_0 . For illustration, we can observe three distinct cases where the antenna polarisation is aligned with the x-axis, y-axis or at a 45-degree angle to both the x and y-axis. In the first case, antenna 1 suffers a high degree of polarisation loss ($S_1 \ll S_2$) which comes from Eq. 5.7.16 where $b_0^2 \ll a_0^2$. This relationship indicates that the source antenna is most likely aligned with the x-axis. The second case is derived in a similar fashion, ($S_1 \gg S_2$) which leads to $b_0^2 \gg a_0^2$. The third case is special, if the source antenna is rotated 45 degrees to both the x and y-axis the polarisation loss in antenna 1 and 2 would be the same ($S_1 = S_2$). This leads to an ambiguity where $b_0 = \pm a_0$. This is a special case and would not likely happen in practice.

It is possible to extend this method to source antennas orientated in all three dimensions by adding another uniquely orientated measuring element to the Multi-copter. This method is also only applicable to linearly polarised source antennas and is sensitive to the accuracy of the initial de-embedding process. Nevertheless, it is possible to scan the beam pattern of an appropriately orientated antenna as well as to de-embed some information regarding its polarisation. For interest, Appendix G demonstrates an antenna beam scanning measurement using the Multi-copter.

5.8 Conclusion

This chapter centred on the development of an aerial measurement vehicle focussed on RFI and propagation measurements. Other RF metrology Multi-copter platforms have always focussed on antenna array characterisation or the measurement of base-station transmitter patterns. The first section started

with the vehicle design that included the decisions made on mechanical performance and flight electronics. This section was followed by an overview of the receiver that discussed the narrow band receiver as well as its calibration used for propagation measurements in a later part of this work. The latter was followed by an investigation into a compact time-domain receiver for broadband measurements. This design was derived from a larger existing architecture called RaTTy. To enable measurements, a novel dual-isotropic antenna system was implemented using integrated-broadband antennas embedded into the landing structure of the vehicle. Antenna patterns were kept stable using a fixed shielded box, encompassing all of the wiring and subsystems. By simplifying the complex environment, full FDTD simulations on the vehicle's antenna performance were made possible. These patterns were finally verified using a near-field scanner. Previously the effect of the complex environment on the on-board antennas were ignored. This made measurements only possible in a relative sense. Our system is capable of absolute measurements by de-embedding the well characterised antenna pattern from the data.

Additionally, the RF signature of the Multi-copter was measured using a reverberation chamber. Here a clean/dirty room shielding scheme is proposed for future work. Also, an algorithm was developed for the de-embedding of measurements from the Multi-copter. This de-embedding procedure formed a core part of the next chapter. The focus in the next chapter will be on full-scale measurements at the KAPB site.

Chapter 6

Propagation and Shielding Measurements

6.1 Introduction

In this Chapter, attention shifts now to full-scale measurements at the KAPB site. These measurements, as mentioned earlier, are expensive and generally focussed. The aim here will be to establish a baseline shielding budget. This budget will allow the user to calculate the amount of shielding available, given the position of an RFI source and receiver. Using this together with regulations, such as South African Radio Astronomy Service (SARAS) protection levels [80], should create a powerful tool to determine the severity of on-site noise sources.

Firstly, the KAPB measurements will be discussed, analysed and used to extract a minimum-shielding level. After that, the berm measurements will be used to create a scattering model, using ITU-R P.526-12 [81]. This will be followed by an empirical model predicting ground loss obtained from open-terrain measurements. Finally, all of these models will be combined to form the shielding budget discussed at the end of this chapter that will be verified by an independent set of measurements.

6.2 Testing the KAPB shielding effectiveness

The KAPB, a power and computing facility for the SKA project, resides at the heart of the site base, see Fig. 6.1. Such a facility, with all of its sub-systems, is expected to generate a substantial amount of noise. Additionally, its location relative to the sensitive KAT-7, Meerkat and SKA receivers, causes its EMC performance to become relevant. It is impractical for the facility only to have noise-free equipment. Therefore, EMC policies needed to be applied throughout the construction phase. A key part of this was to use the building rebar structure as a Faraday cage and its concrete as an absorbing



Figure 6.1: Contemporary photo of the KAPB, dark roof, taken from the top of Losberg. The berm and the assembly shed can be seen on the left and right of the KAPB respectively. Photo was taken by Dr P. G. Wiid.

material. Together these factors limit the amount of energy that could escape the building. The shielding factor describes the measure of signal reduction through a barrier. It should be noted that although the KAPB is located at the heart of the site, it is still located on the opposite side of a Losberg in relation to the SKA core. Losberg is a flat-top hill 300 m high and 3 km wide. Additional to the shielding Losberg provides, the core of SKA is still located several kilometres further from Losberg. Therefore shielding to the core site is sufficient, this study will focus on receivers out of the core that are not directly shielded by Losberg.

6.2.1 Polar pattern measurement

In the shielding analysis undertaken in this section a transmitting antenna is placed, at various locations, inside the KAPB building. A measurement, $S_m(x, r)$, is then made in a polar-pattern shape, on the outside of the building at points x . The antenna is then moved to another room in the building, r , as a reverberation mechanism. Subsequently, the measurements are repeated for the new transmitting location. A free-space predicted value, $S_p(x, r)$, is then calculated to each measured point outside the building using Eq. 6.2.1. The difference between the latter predicted and measured values forms the

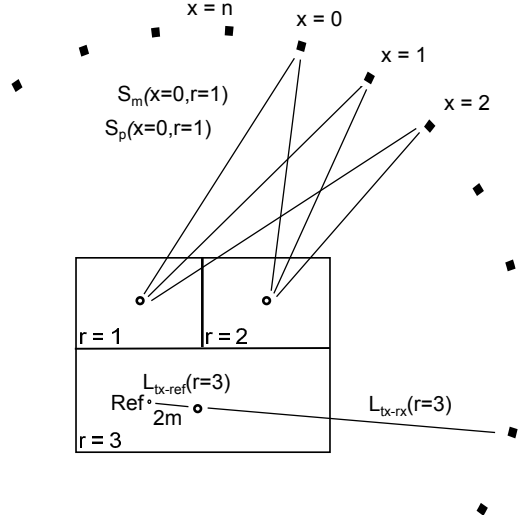


Figure 6.2: Diagram illustrating how shielding, S , is calculated around the KAPB, at points x , while transmitting from different rooms, r , inside the building.

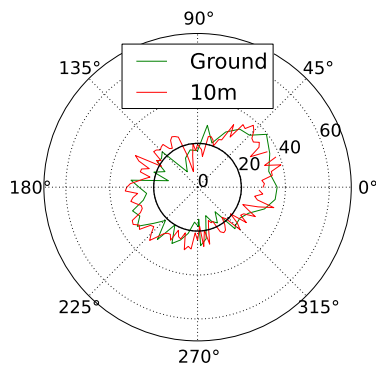
shielding factor, see Eq. 6.2.3. In post-processing, the minimum-measured shielding for each transmitting position is used as the shielding factor for that polar position. $S_r(r)$ is the measured reference level of the antenna, 2 m away, inside the specific room. $L_{tx-rx}(x, r)$, in Eq. 6.2.2 is the free-space loss from the transmitting position to the measurement position. $L_{tx-ref}(x, r)$ is the free-space loss from the transmitting position to the reference position, 2 m in front of the antenna. Fig. 6.2 gives a diagrammatic overview of the measurement and the variables mentioned.

$$S_p(x, r) = S_r(r) - (L_{tx-rx}(x, r) - L_{tx-ref}(x, r)) \quad (6.2.1)$$

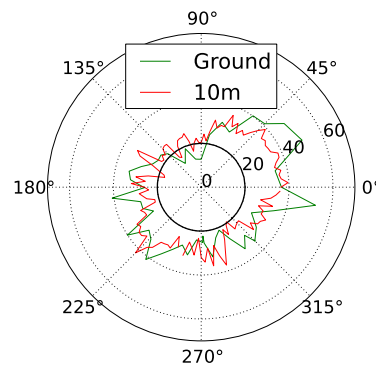
$$L = 20 \log_{10}(d) + 20 \log_{10}(f) - 27.55 \quad (6.2.2)$$

$$S_x = \min(S_p(x, r) - S_m(x, r)) \quad (6.2.3)$$

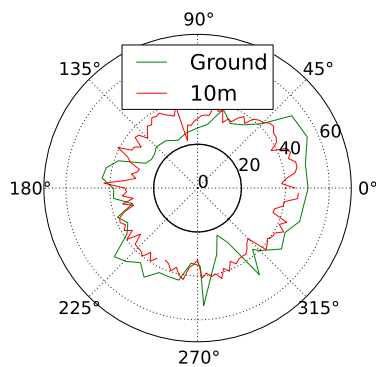
After calculating shielding polar-patterns for each transmitting position, it is possible to extract the overall shielding of the building. Fig 6.3 shows the measurement of the Multi-copter, at three discrete frequencies, compared to those taken at ground level (Thanks to MESA [82] for sharing their ground-level measurements). The vehicle measurements were all done at a height of 10 m. At this height, the receiver is in the direct line from the building to the top edge of the berm. It can be seen with both trends that at 0 and 180 degrees the shielding is higher than at 90 and 270 degrees. This means that at the critical direction, the direction of the berm and outlying receivers, the shielding is far more than the 20 dB level. This level has been arbitrarily



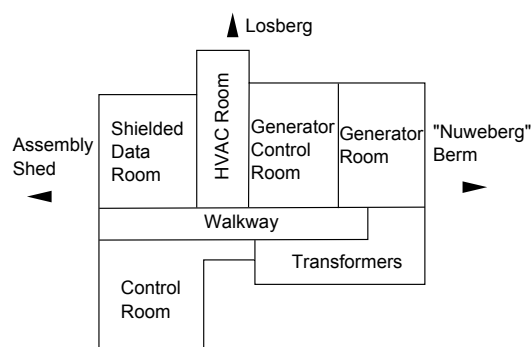
(a) Measurement at 290 MHz



(b) Measurement at 430 MHz



(c) Measurement at 905 MHz



(d) Orientation of building

Figure 6.3: Shielding measurement of the KAPB at three discrete frequencies. Ground measurements were done using a hand-held spectrum analyser while measurements at 10 m height were accomplished using the Multi-copter platform. The measurements can be compared to the diagram in (d) for a sense of orientation.

chosen to add perspective to the plots. During the final campaign, it was found that the top of the assembly-shed structure is higher than the top of the berm. This could cause the shed, which is a metallic structure, to act as a reflector over the berm. Therefore, it is also necessary for the shielding at 180 degrees to be at acceptable levels. The lower shielding at 90 and 270 is related to apertures that have been identified on the site. Measures can be put in place to reduce these apertures should a need to increase shielding in these directions exist. At present, there is no requirement for extra measures, and the inherent policies are successful. It also becomes apparent that, as frequency increases, the overall shielding increases. The latter is attributed to the absorbing properties of the concrete structure.

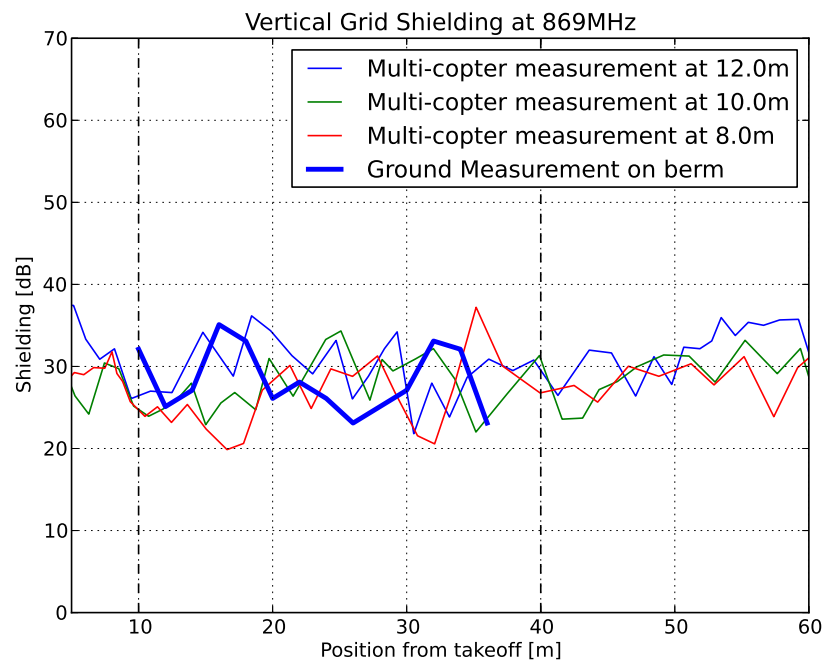


Figure 6.4: Shielding measurement along the berm-side of the KAPB compared to ground measurements on top of the berm. Vertical lines indicate the start and stop of the KAPB berm-side wall.

6.2.2 Earlier campaign shielding measurements

These final measurements can be compared to a previous campaign when construction was still active. In these measurements, the Multi-copter was flown in a vertical-grid pattern on the berm side of the KAPB. The vehicle travelled in straight lines from the North side to the South side of the building, at equally spaced vertical heights. The shielding level at 10 m can be seen in Fig. 6.4. Here the average shielding is lower. This shows that the grounding and cabling policies implemented, subsequent to this campaign, improved the shielding. The full vertical-plane measurement can be seen in Fig. 6.5. The height axis of this measurement is sampled in 2 m steps. Because of this sparsity, a linear interpolation has been applied to the dataset. It would be possible to measure height levels within fractions of a wavelength. This will however increase the measurement time, which would have been impractical during this campaign.

To investigate how shielding changes with height, measurements were taken along vertical lines on each side of the building. The vertical-shielding measurement, for the berm side of the KAPB, can be seen in Fig 6.6. It is evident from this that shielding tends to decrease with increasing height. This is rather intuitive and can be understood by examining the diagram in Fig. 6.6b. At lower heights, the receiver is shielded by the ground, as well as multiple verti-

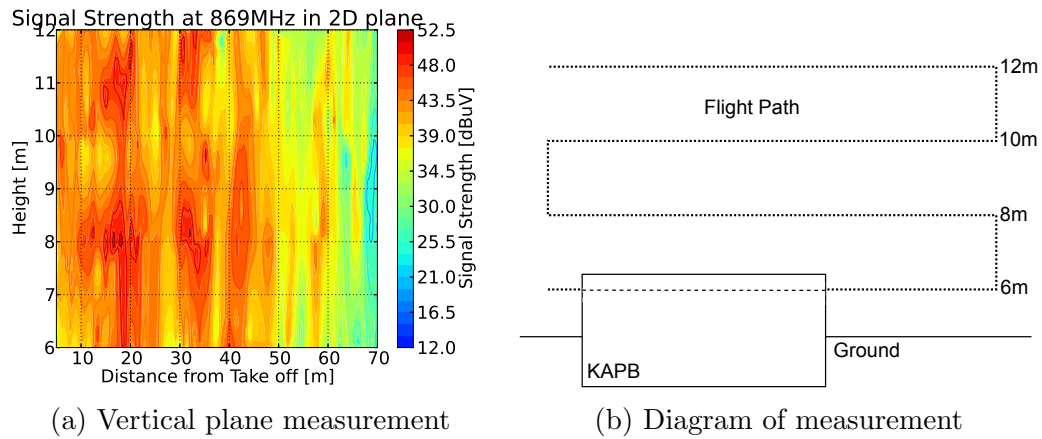


Figure 6.5: Vertical-plane signal strength graph, measured between the KAPB and berm. The diagram in (b) gives an overview of how the measurement was planned.

cal walls. However, as height increases the roof of the building will, in the end, be the only shielding structure. This will not affect the shielding performance of the site as emissions should propagate over the berm without diffracting. Consequently, these emissions should not affect nearby receivers. The shielding at 10 m and below is of most concern. Comparing these trends to the polar-shielding graphs in Fig. 6.3, the decreasing trend, between ground level and 10 m, is not as pronounced. This again shows how the shielding effectiveness has increased between the two measurement campaigns. The shielding graph was calculated using two different transmitting locations. Measurements from these locations were then transformed, as discussed in Section 6.2.1 into shielding data. After this, the minimum shielding, from the two transmitting locations, at each point is used as the total shielding estimate. The individually measured positions can be found in Appendix H.

6.2.3 KAPB shielding estimation

Using the latest campaign data in Fig. 6.3, a minimum shielding value can be extracted. This value should represent the worst-case shielding scenario. Barring material and structural resonances will be assumed to be predictably frequency-dependent. To simplify the shielding model, only the berm-side, -45 to 45 degrees, shielding will be used. The minimum values for each frequency can be seen in Table 6.1. These values are read from the polar graphs, taking the minimum of both the ground as well as the 10 m measurement into account. With these three tabulated points, a quadratic fit was used to create an analytical model for the minimum shielding. This model is only valid towards the berm side of the KAPB. The fitted model can be seen in Eq. 6.2.4 and plotted in Fig. 6.7. Here shielding, $L_{KAPB}(f)$, is represented in dB. A

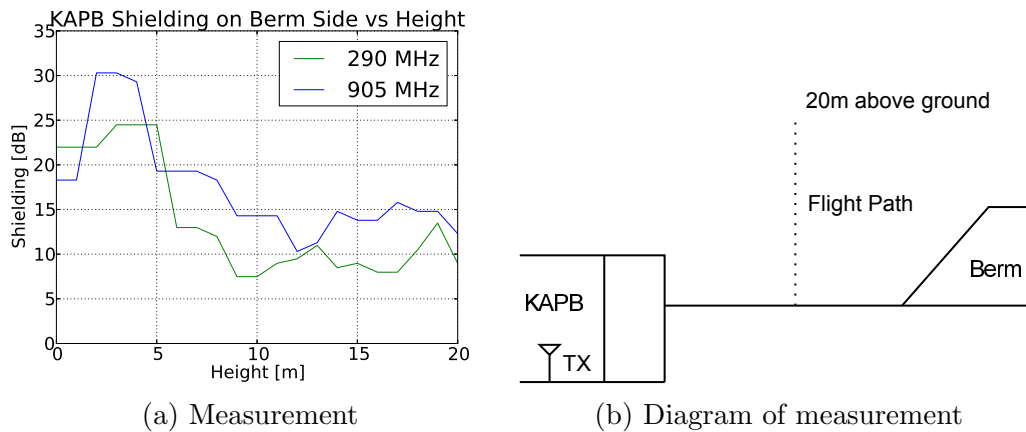


Figure 6.6: Vertical shielding measurement, on the berm side of KAPB. This plot shows, at two discrete frequencies, how shielding tends to decrease with height. A diagram showing how the measurement was conducted can be seen in (b).

Frequency [MHz]	Shielding [dB]
290	22.5
430	24.0
905	36.5

Table 6.1: Minimum shielding levels on the berm side of KAPB extracted from Fig. 6.3.

quadratic fit was used to depict accurately the three measured values and also underestimate the shielding in between. The quadratic fit is however only valid between the measured frequencies.

$$L_{KAPB}(f) = 2.54 \times 10^{-5} f^2 - 7.55 \times 10^{-3} f + 22.56 \quad (6.2.4)$$

where

- f is frequency in MHz.

The purpose of this modelling is to facilitate a means to estimate the KAPB shielding over a reasonable frequency range. This could be used to predict noise levels outside of the building for a given interferer's emission levels within the building. The next step is now to obtain another empirical model, based on campaign measurements, for ground propagation.

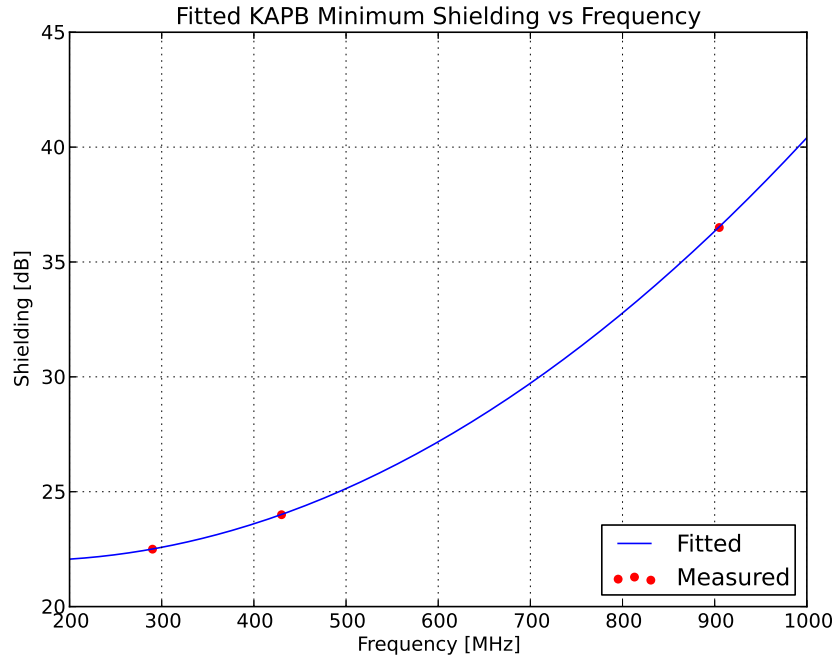


Figure 6.7: Minimum shielding vs frequency trend extracted from polar pattern shielding graphs. A square fit was used to approximate the measurements, which can now be used as an analytical tool.

6.3 Ground loss

The RFI from noise sources at the site base will be further attenuated by the presence of ground. This attenuation will be additional to free-space. There are numerous models available; most of these can be found in the ITU-R P1546 [41] and ITU-R P452 [83]. These include empirical models such as Young, Okumura and Hata as well as models using terrain maps such as Longley and Rice [22]. For the ground loss, two simple models known as the Elgi [39] model and the Two-ray ground reflect model will be used. These models are preferred in this situation as they do not require terrain models. The equation for the Egli and Two-ray model can be seen in Eq. 6.3.1 and 6.3.2 respectively. The two-ray model is defined with a dual slope, where Eq. 6.3.2 is only valid from a critical distance $d_c = \frac{4\pi h_s h_r}{\lambda}$. Before this point free-space loss should be used.

$$L_{Ground} = L_{50} = 10 \log_{10} \left(\left(\frac{h_s h_r}{d^2} \right)^2 \left(\frac{40}{f} \right)^2 \right) \quad (6.3.1)$$

where

- h_s is the height of the RFI source in metres

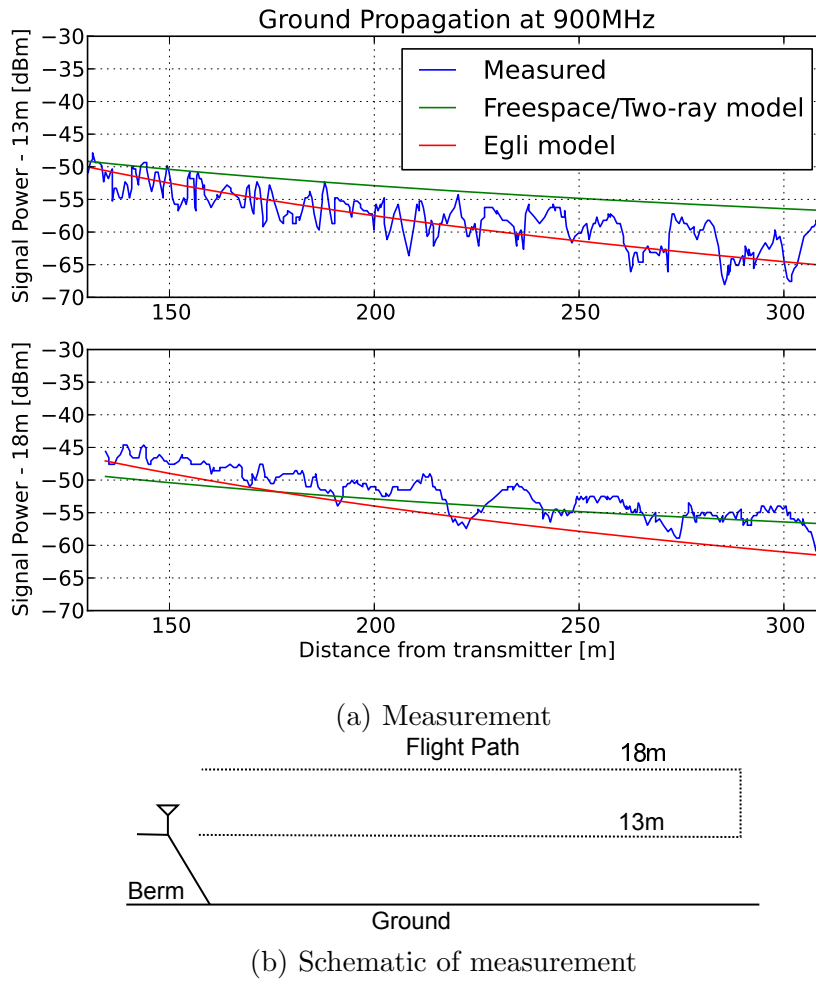


Figure 6.8: Ground loss measurement compared to the simple ground loss model at two different heights. A free-space prediction can also be seen for comparison. A schematic in (b) gives an overview of the measurement setup.

- h_r is the height of the receiver in metres
- d is the distance between the RFI source and the receiver in metres
- f is the frequency in MHz

$$L_{Ground} = -40 \log_{10}(d) + 10 \log_{10}(h_s^2 h_r^2) \quad (6.3.2)$$

To validate these models at the Karoo site, measurements were taken using the Multi-copter over relatively-open terrain. The antenna was placed on the berm, with the vehicle flying away at a height of 10 m and 15 m from the ground (see Fig. 6.8b). The idea here is to eliminate most terrain obstacles and simulate a flat earth environment. The measurement compared to the

models can be seen in Fig. 6.8. Note that the critical distance for the Two ray model is beyond 340 m at these frequencies. Therefore, these comparisons focus primarily on the Egli and free-space loss models. The two-ray model will be compared to measurements above its critical distance in Section 6.7 as part of the full-model verification. It can be seen that the Egli model closely follows the average trend of the measurement. The variation around the average is attributed to the proximity of Losberg as well as the effect of the berm below the transmitter. The free-space/two-ray based prediction can also be seen which demonstrates how the presence of ground increases the propagation loss. The latter analytical model underestimates the shielding and will be useful as a minimum shielding projection. This was a relatively short range measurement. Therefore, effects such as gas attenuation, foliage, troposcatter and refraction were not taken into account. For the scope of this work, the focus is mainly on the local propagation where these effects are negligible.

6.4 Characterisation of the berm

The berm is a result of a strategic decision to dispose of the excavated soil in a manner that would benefit the shielding properties of the site. It is for this reason that the man-made structure has been placed next to the KAPB. The idea of the berm is to absorb and reflect RFI propagating from the building. This section will investigate the extent to which the berm shields the nearest receivers. Additionally, a model will be created that can be used to predict the shielding at arbitrary locations. A diagram detailing the structure of the berm can be seen in Fig. 6.9.

6.4.1 Diffraction measured over berm

A primary factor determining the shielding of the berm is its shape, such as the sharpness of its edges. Such edges cause signals, incident on the berm edges, to diffract. This diffracted energy will then radiate, with a lobed antenna pattern, over the berm. Depending on the position of the receiver, this

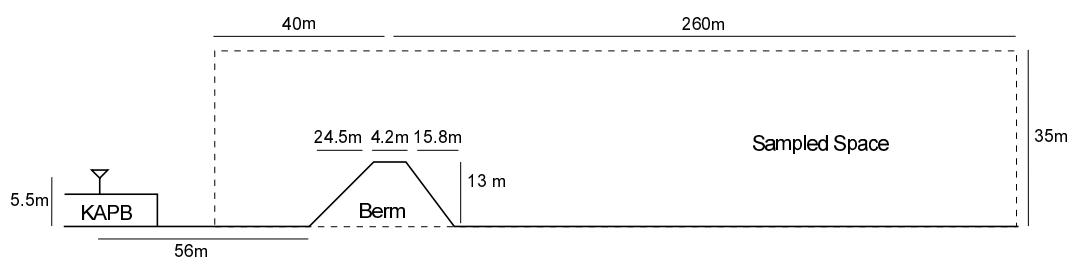
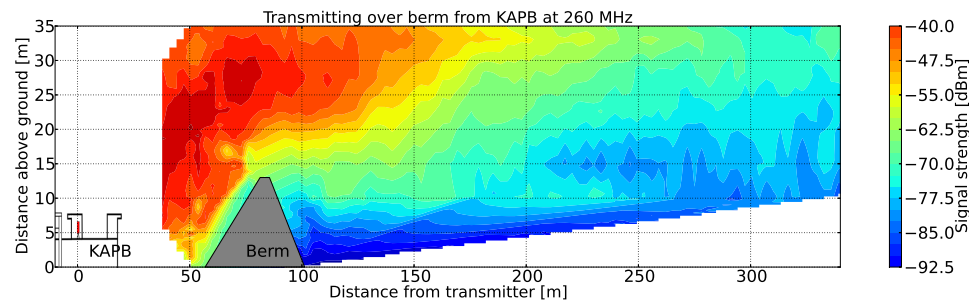
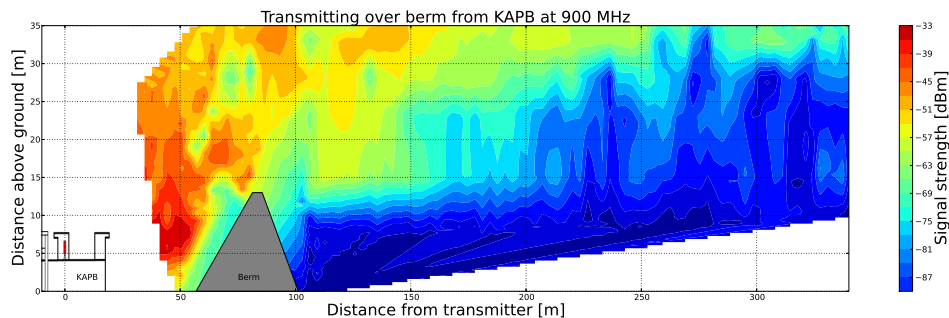


Figure 6.9: A diagram of the berm and KAPB environment. Here the location of the antenna can be seen. A region called sampled space was measured to obtain the vertical-plane measurements in Fig. 6.10.



(a) Measurement at 260 MHz



(b) Measurement at 900 MHz

Figure 6.10: Vertical plane measurements of a noise source transmitting over the berm. At the two discrete frequencies diffraction is seen in the shadowed region of the berm.

antenna-pattern characteristic might increase or decrease the shielding effectiveness of the berm. To illustrate this behaviour, a vertical 2D measurement was taken in-front, over and behind the berm. Here a bi-conical antenna was placed on the roof of the KAPB, simulating a radiating noise source, transmitting at 19 dBm. This measurement consisted of multiple horizontal flights at 5 m incrementing height levels. The data was then compiled into a single set and interpolated to provide the results in Fig. 6.10. All of the measurements were within 1 metre of the reference plane. It is clear in these measurements that a diffraction mechanism is present. This phenomenon is seen as the increased signal strength measured in the area shadowed by the berm. Additionally, a diagram explaining the measurement configuration can be seen in figure 6.9.

The berm is not unique, and soil structures such as these have been analysed in the past. There exists literature, such as ITU-R P.526-12, which contains diffraction models for these kinds of terrestrial obstacles. These models make use of a single, or multiple finitely conducting wedges to approximate the structure, depending on its shape. During the final measurement campaign, an effort was made to measure berm diffraction patterns. In these measurements the Multi-copter was flown, at a single GPS point, to an altitude of 100 m. This measurement point is located on the far side of the berm and transmitter, sim-

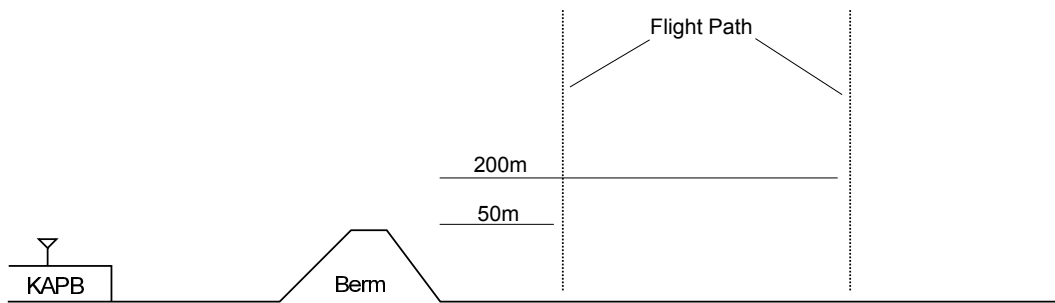


Figure 6.11: A diagram of the diffraction measurements. Here the two measurement locations along with the transmitting position, on the far side of the berm, can be seen. Three discrete frequencies were measured at each location.

ilar to the vertical-grid measurement. A total of three discrete frequencies for two positions, at different distances from the berm, were measured. A diagram of the measurements can be seen in Fig. 6.11. These measurements will be used to investigate and test the best model configuration. This configuration will then be tested against the rest of the measurements.

6.4.2 Diffraction models

The most simple diffraction model is a knife-edge. Depending on the application, this model can be expanded into multiple knife-edges, rounded knife-edges or even finitely-conducting wedge-shape diffractors. Geometries for these models can be seen in Fig. 6.12. Because of the relatively small distance between the transmitter and receiver, as well as the dynamic nature of the diffracting propagation, it was decided not to integrate the ground loss into these measurements. The ground loss at these distances is very much comparable to a free-space approximation. These models will now be simulated for a fixed transmitter and a moving receiver in the manner illustrated in Fig 6.11.

The primary inputs to the diffraction models were chosen to be similar to the dimensions of the berm. Therefore, the height of the knife-edge and wedges are chosen as the height of the berm. Similarly, the width of the berm was used to define the angle of the wedge. In the case of the double configurations, the top width of the berm was used as the spacing between the two diffractors. Material parameters of the Karoo soil [59] were also taken into account in the wedge models. An initial comparison between the models and a measurement can be seen in Fig 6.13. In this comparison, it is clear that the knife-edge models closely approximate their double counterparts. From this initial comparison as well as a further parameter study, it was found that the double configuration overall does not model the measurement as well as the single version. However, the oscillations present above 50 m are always better explained by the double diffractor, which is not the case with the single diffractor. For the purpose of this study the initial trend, at lower altitudes,

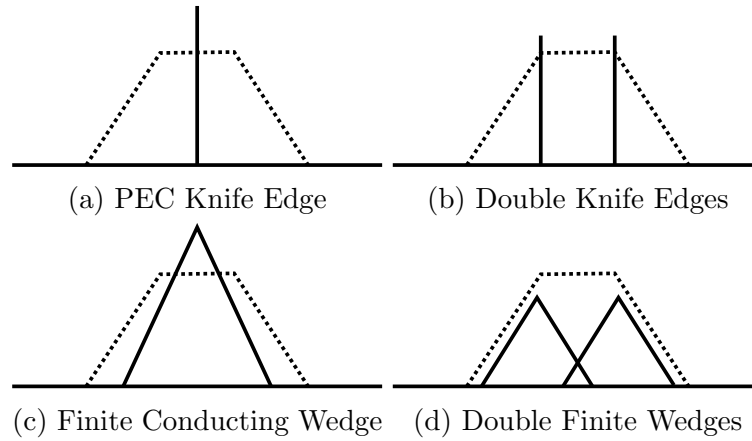


Figure 6.12: Different diffraction model geometries used in the characterisation of the berm. These models range from infinitely-thin PEC knife edges to finitely conducting wedges.

were of greater importance. For this reason, the single diffractor will be used as the model of choice. Also, it was decided to use the knife-edge diffractor to minimise the amount of computation.

With a suitable model, it is now necessary to calibrate and test its predictions to empirical data. The knife-edge model takes into account three

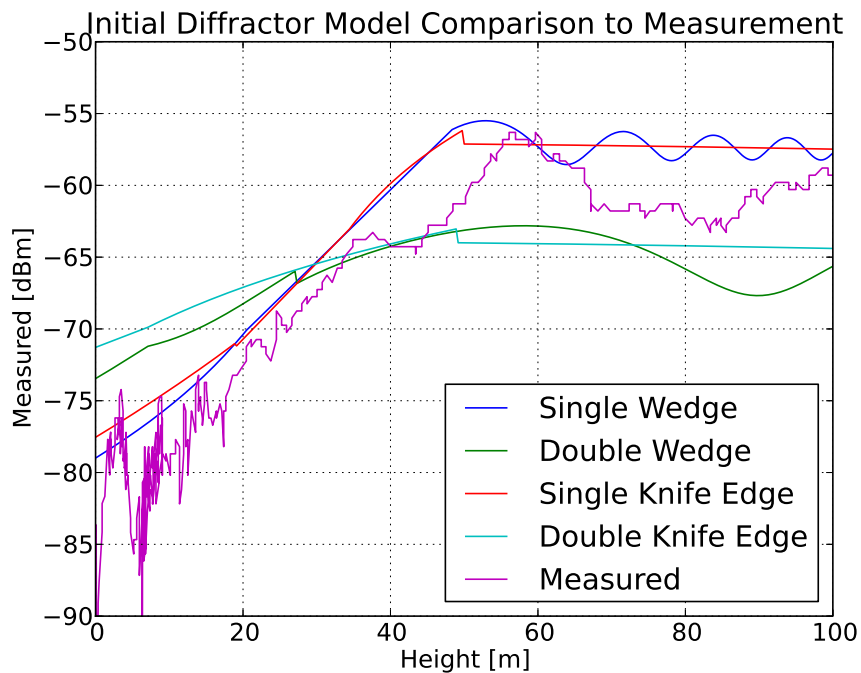


Figure 6.13: Comparison between real measurement and model configurations seen in Fig 6.12. Measurement was taken at 550 MHz 200 m behind berm.

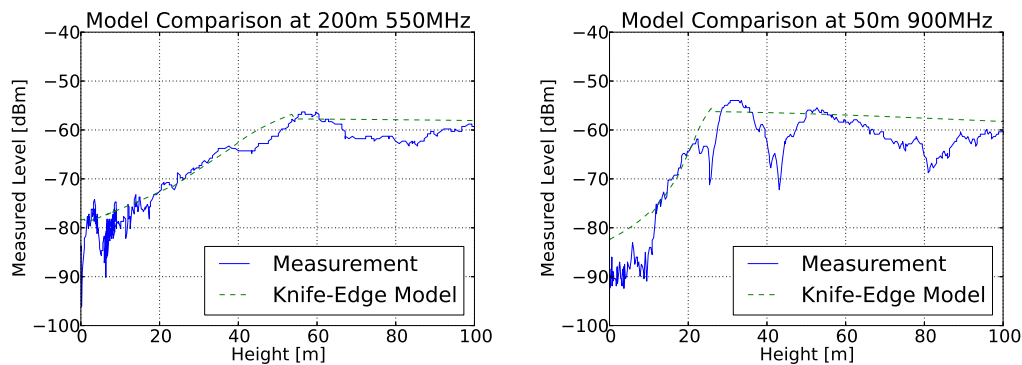


Figure 6.14: Comparisons between measurement and calibrated model for berm diffraction. These graphs represent two different positions as well as two different frequencies behind the berm.

parameters: frequency, the height of the transmitter, the height of the receiver and the relative height of the diffractor above LOS. The knife-edge itself is an oversimplification of the berm structure. Therefore, the height of the berm would not necessarily be the same for a knife-edge to produce the same diffraction. In this case a higher knife-edge will serve as a better prediction. Consequently, an optimisation was used to calibrate the knife-edge parameter. This optimisation minimised the error between the measurement and prediction. It should also be noted that the optimisation was weighted to lower heights. The latter decision reduced the effect of the knife-edge approximation on the optimisation. The final optimised height value for the knife-edge was found to be 14 m (actual berm height is 13 m). This height parameter will be constant for all frequencies and receiver locations. The agreement between the subset of the diffractor measurements and their respective predictions can be seen in Fig. 6.14. The rest of the comparisons can be found in Appendix H.

From the comparisons, it is clear that the knife-edge model works. The model tends to underestimate the shielding of the berm, which is needed for a minimum shielding calculation. It was also found that at the lower frequencies (260 MHz) the measurements are about 10 dB lower than the model as well as the free-space loss section. It is argued that the proximity of the building to the transmitting antenna might have loaded the antenna at this low frequency. This trend does not carry through to the higher frequencies.

The loss for the berm (knife-edge model) is calculated using the approximated diffraction integral in Eq. 6.4.1 from Lee [84]. This equation can now be used to determine the loss caused by the berm to any distant point within

the flat-terrain region.

$$\begin{aligned}
 L_{Berm} &= 0 & v &\leq -1 \\
 L_{Berm} &= 20 \log_{10}(0.5 - 0.62v) & -1 &< v \leq 0 \\
 L_{Berm} &= 20 \log_{10}(0.5e^{-0.95v}) & 0 &< v \leq 1 \\
 L_{Berm} &= 20 \log_{10}(0.4 - \sqrt{0.1184 - (0.38 - 0.1v)^2}) & 1 &< v \leq 2.4 \\
 L_{Berm} &= 20 \log_{10}\left(\frac{0.225}{v}\right) & v &> 2.4
 \end{aligned} \tag{6.4.1}$$

where

- v is $h\sqrt{\frac{2(d_1+d_2)}{\lambda d_1 d_2}}$
- h height of the knife-edge above the LOS line in m
- d_1 distance between the source and the top of the knife-edge in m
- d_2 distance between the receiver and the top of the knife-edge in m
- λ wavelength of the RFI in m
- L_{Berm} is the modelled loss caused by the berm in dB

6.5 Site-base shielding budget

With the most of the local propagation aspects modelled, it is now possible to determine a shielding budget for on-site RFI. The shielding budget will function as a tool to determine if a noise-source is harmful to the nearest receiver. This will be done using the SARAS levels seen in Eq. 6.5.1 as a reference. The calculations for this example will be done to the nearest, out of the core, MeerKAT-receiver not shielded by Losberg (See Fig. 6.15).

$$SARAS = -17.2708 \log_{10}(f) - 192.0714 \tag{6.5.1}$$

where

- f is the frequency in MHz, less than 2000 MHz
- $SARAS$ is the maximum tolerable level in dBm/Hz

Using the models in Eq. 6.2.4, 6.3.1 and 6.4.1 a shielding level can be calculated from either inside the KAPB, outside the KAPB and outside the KAPB not shielded by the berm. This is possible by just changing the way the equations are handled. The shielding levels for each of the three mentioned scenarios can be seen, as an example, in Fig. 6.16. In the "outside KAPB" case

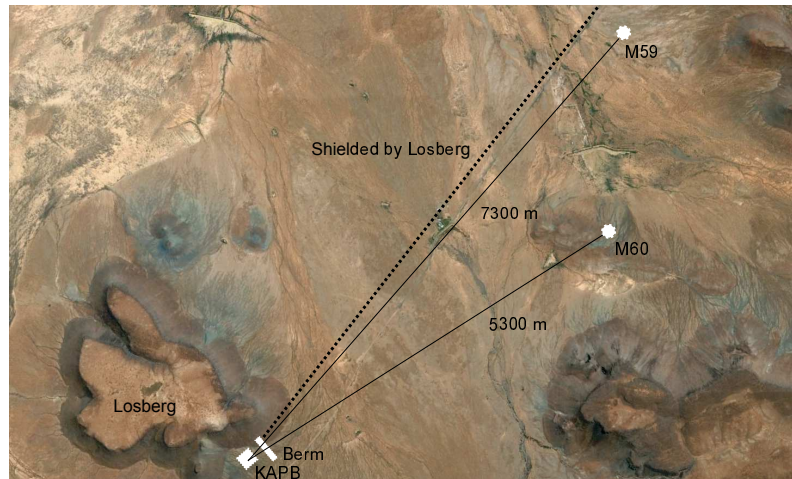


Figure 6.15: Google Earth satellite map showing the location of the KAPB and berm relative to the nearest MeerKAT receiver.

the RFI is placed close to the living quarters at a height of 2 m to simulate the use of a laptop. Similarly in the "Not shielded by berm" case the RFI is moved into the field region removing the berm obstruction. Here the distance and height is kept the same as the previous case. With this prediction, it can be seen that RFI inside the KAPB has at least 150 dB (200 MHz) shielding to the closest MeerKAT receiver. This shielding represents an unshielded RFI source inside the building. Equipment placed in a shielded compartment would benefit from shielding additional to that calculated in these models. Assuming an RFI source with a bandwidth of 12.5 kHz, the maximum allowable signal, according to SARAS, can be seen in Fig. 6.16b for a source located inside the KAPB.

A step-by-step procedure for obtaining the overall shielding as in the examples will be explained. It should first be noted that these models were only tested between 260 MHz and 930MHz. Also, a noise source inside the KAPB is assumed to radiate, after shielding has been applied, at a height of 5.5 m from the centre of the KAPB. Berm shielding is assumed for propagation perpendicularly over its centre. In the case where berm shielding is applied, the source height for the ground-loss equation is taken as the height of the berm where the ground is approximated as a flat earth. Finally, propagation from outside the KAPB does not take into account obstacles such as containers, sheds and other on-site buildings.

- **Step 1:** If the RFI source is inside the KAPB continue to Step 2, otherwise take $L_{KAPB} = 0$ and skip to step 6.
- **Step 2:** Calculate the shielding L_{KAPB} delivered in the direction of the berm using Eq. 6.2.4.

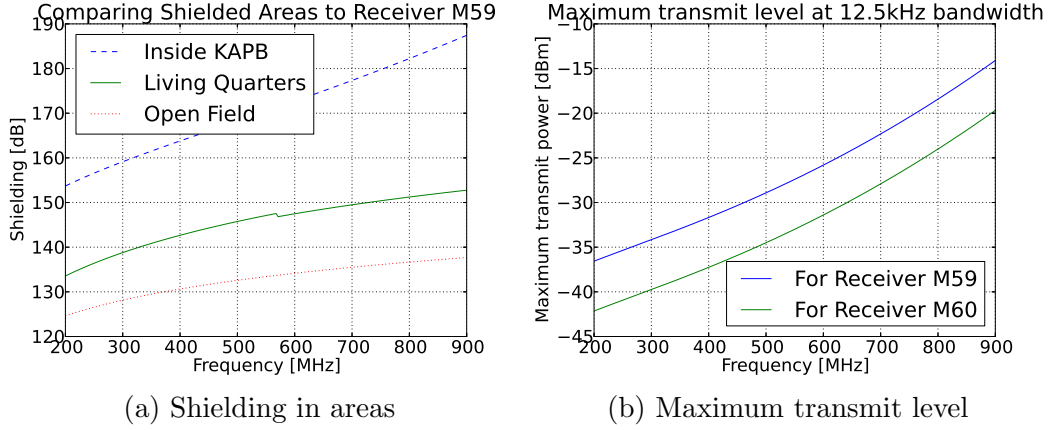


Figure 6.16: Shielding to nearest MeerKAT receiver (M59) from inside KAPB, outside KAPB and outside KAPB not shielded by the berm. Height for the outside RFI positions were kept at 2 m high 70 m from the foot of the berm. In (b) the maximum allowable signal transmitted, from inside the KAPB, can be seen graphed using (a).

- **Step 3:** Calculate the berm shielding L_{Berm} with Eq. 6.4.1. Here v (Eq. 6.5.5) is calculated using a simplified form of v in Eq. 6.4.1.

$$v = \frac{1.34x_r - 12.62h_r + 176.62}{x_r + 80} \sqrt{\frac{80.45 + d_2}{\lambda d_2}} \quad (6.5.2)$$

where

- λ is the wavelength of the RFI
- h_r is the height of the receiver above ground
- $x_r = x_{tot} - 80$ which is now the distance from the berm to the receiver
- x_{tot} is the distance from the KAPB to the receiver
- $d_2 = \sqrt{(x_r)^2 + (14 - h_r)^2}$
- d_2 can be approximated as x_r when $x_r \gg 14$

- **Step 4:** Calculate the ground loss L_{Ground} using Eq. 6.5.3 which is a simplified form of Eq. 6.3.1.

$$L_{Ground} = L_{50} = 10 \log_{10} \left(\left(\frac{13h_r}{(x_{tot} - 80)^2} \right)^2 \left(\frac{40}{f} \right)^2 \right) \quad (6.5.3)$$

where

- f is the frequency of the RFI

The two-ray model (Eq. 6.3.2) can also be used for a worst-case shielding scenario leading to Eq. 6.5.4. Keep in mind that the latter equation is only valid for distances beyond the two-ray model critical distance.

$$L_{Ground} = 10 \log_{10} \left(\frac{13h_r}{(x_{tot} - 80)^2} \right)^2 \quad (6.5.4)$$

- **Step 5:** Continue to step 11
- **Step 6:** If the Berm is located between the RFI source and the receiver continue to the next step, otherwise take $L_{Berm} = 0$ and skip to step 10
- **Step 7:** Calculate the berm shielding L_{Berm} with Eq. 6.4.1. Here v (Eq. 6.5.5) is calculated using a simplified form of v in Eq. 6.4.1 with variable source position and height.

$$v = \frac{x_r(14 - h_s) - x_s(14 - h_r)}{x_r - x_s} \sqrt{\frac{2(d_1 + d_2)}{\lambda d_1 d_2}} \quad (6.5.5)$$

where

- h_s is the height of the RFI source from the ground
- $x_s = -(x_{sb} + 20)$
- x_{sb} is the distance from the RFI source to the foot of the berm
- $d_1 = \sqrt{(x_s)^2 + (14 - h_s)^2}$
- **Step 8:** Calculate the ground loss L_{Ground} in the same manner as in Step 4.
- **Step 9:** Continue to step 11.
- **Step 10:** Calculate the ground loss L_{Ground} using Eq. 6.5.6 which is similar to Eq. 6.3.1.

$$L_{Ground} = 10 \log_{10} \left(\left(\frac{h_s h_r}{x_{tot}^2} \right)^2 \left(\frac{40}{f} \right)^2 \right) \quad (6.5.6)$$

Or use the two-ray model in Eq. 6.5.7 for the worst-case shielding scenario. Keep in mind that the latter equation is only valid for distances beyond the two-ray model critical distance.

$$L_{Ground} = 10 \log_{10} \left(\frac{h_s h_r}{x_{tot}^2} \right)^2 \quad (6.5.7)$$

- **Step 11:** Calculate the total shielding using Eq. 6.5.8.

$$S_{total} = L_{KAPB} + L_{Berm} + L_{Ground} \quad (6.5.8)$$

- **Step 12:** Calculate the SARAS level P_{Saras} for the given frequency from Eq. 6.5.1.
- **Step 23:** Calculate the maximum transmit level P_{max} using Eq. 6.5.9.

$$P_{max} = P_{Saras} + S_{total} \quad (6.5.9)$$

Using these empirically-based models along with the steps laid out in this section, it is now possible to calculate the amount of buffer available between the on-site noise sources and the SARAS reference levels. For an on-site RFI Engineer, this could be used as a tool to determine the severity of RFI sources. Therefore, sources exceeding the maximum transmit level can be flagged. While sources well below the maximum transmit level can be left in place to save resources.

6.6 Comparing measurements to the computational model

In the previous section, a set of simple propagation models was proposed. These models should suffice for most applications. However, in the case where a more complex task is at hand the computational model developed earlier can be used. This could include the investigation of a certain terrestrial feature or possibly its design. As a rigorous test, the computational model can be tested to the real-world measurements. Here the same FEKO[®] model used in Section 4.6 is configured to represent the latest campaign measurements. A comparison between a diffraction flight and its simulation can be seen in Fig. 6.17. The calibrated knife-edge diffraction model developed earlier and a 2D boundary integral model¹ has also been added to the comparison. This illustrates how the computational model can also be used as means of developing simplified propagation models for field use. It is evident that the simulation closely represents the measured data except for a set of nulls above 60 m. These nulls are caused by the de-constructive reflections from the PEC-ground plane. Nevertheless, data where at least one of the mirrored sources are shadowed by the berm are not as severely affected. Because the interest here is to determine worst-case scenarios, the computational model is deemed fit.

The height at which the reflection becomes visible can be calculated by Eq. 6.6.1.

¹Developed by N. S. Tezel for efficiently optimising the 2D profile of a berm [71]

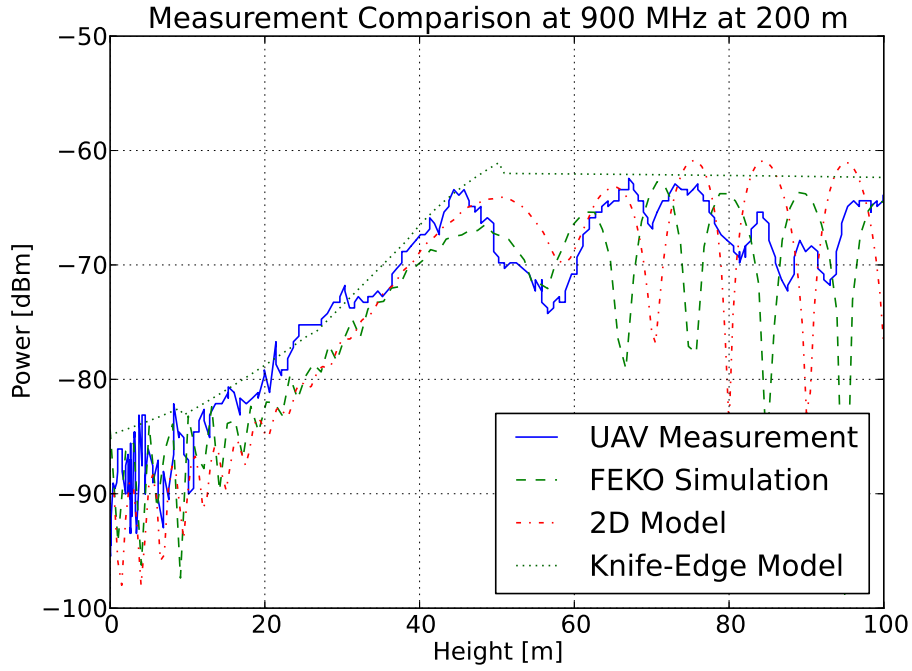


Figure 6.17: Comparison between the computational models and Multi-copter measurements for a diffraction flight 200 m behind berm at 900 MHz.

$$h_n = l_f \left(\frac{h_a + h_s}{l_s} \right) - h_a \quad (6.6.1)$$

where

- h_n is height above ground at which the PEC reflections become significant
- l_f is the distance from the antenna to the flight path
- l_s is the distance from the antenna to the berm structure
- h_a is the height of the antenna above ground
- h_s is the height of the berm structure

This equation calculates the point at which the image source becomes directly visible from behind the berm. In this equation, the berm has been approximated as a simple thin wall to simplify the calculation. The actual geometry of the berm will become more important as the source becomes closer to the berm. Fig. 6.18 shows the geometry of the problem. In the case where the source is electrically distant from the investigated area the effect of these reflections will become less. In this case, the mirrored current sources approach

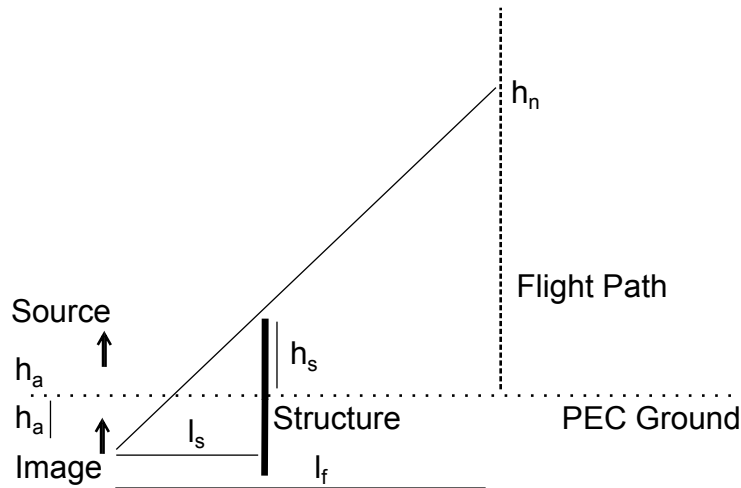


Figure 6.18: Geometry used to calculate when the PEC reflections affect the simulation.

the case of a monopole above ground. In this instance, there are no hard reflections from the PEC ground plane. It should also be noted that in most cases only areas that are out of the reflected region need to be investigated.

6.7 Verification of the simplified propagation model

In Sec. 6.5 a simplified propagation model was developed. The model enables the prediction of shielding effectiveness (SE) between the local KAPB area and nearest telescope receivers. This section focusses on the verification of the model using a set of independent measurements. These measurements were made using a directive antenna and portable battery-powered spectrum analyser. The source antenna, also directive, was connected to a battery-powered signal generator set to 18 dBm. The boresight gain of the transmitting and receiving antennas were 5.05 dB and 17.19 dB respectively at 900 MHz. During measurement, the spectrum analyser was set to maximum hold. The transmitting locations were chosen to cover cases where the noise source was shielded as well as not shielded by the berm.

Using the steps in Sec. 6.5 the shielding for each measured transmitter-shielding pair is calculated. Using the known transmitter level, the measurements can be compared to the model predictions as seen in Fig. 6.19. Both the Egli and two-ray ground-loss models were graphed for each scenario. The Egli model can be seen to overestimate the shielding compared to the two-ray model that tends to be more conservative. This overestimation observed in the Egli model is less prevalent when the transmitter and the receiver are further away from the ground. During these measurements, the transmitter

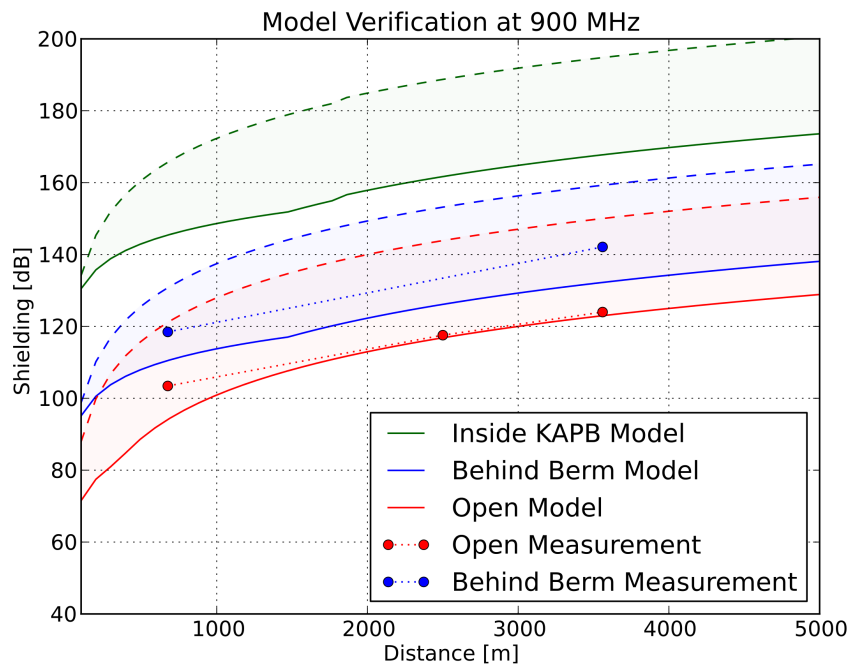


Figure 6.19: Comparison between model predictions and on-site measured data for the three main scenarios. Dashed and solid lines respectively illustrate the difference between using the two-ray or Egli model for ground propagation.

and receiver were both 2.5 m above the ground. These two models give a lower and upper bound to the possible ground loss. The use of either model depends on the application. It is also important to note the dual-curve behaviour of the two-ray model. Below a critical distance, the two-ray model changes to a simple free-space model. At 900 MHz the critical distance is around 400 m. All MeerKAT receivers are located further than this, making it possible only to concentrate on the second part of the two-ray model curve seen in Eq. 6.3.2. From these measurements, it is clear that an upper and lower bound on the main shielding scenarios to nearby receivers has been set. This simplified propagation model is now verified and can be used for on-site RFI severity investigations.

6.7.1 Case study: Power control room emissions in KAPB

To demonstrate the usefulness of the model, a real RFI emission source in the KAPB is investigated. A recent RFI on-site measurement campaign measured the emissions in each room of the KAPB. It becomes necessary to quantify the severity of these measured emission levels on the operation of the nearest MeerKAT receivers. This is done by subtracting the propagation loss from

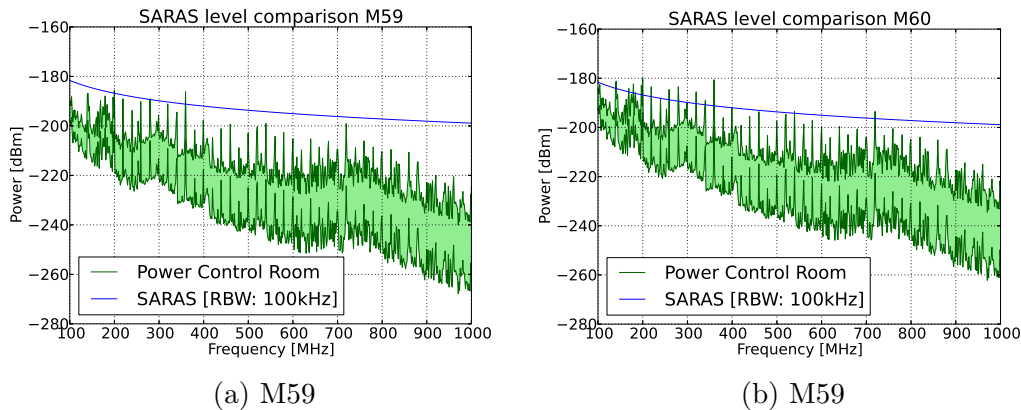


Figure 6.20: Received level at M59 and M60 receivers using the measured emissions in the KAPB, translated using the developed empirical models. An upper and lower bound for shielding were calculated using the models from this work. SARAS levels are added to assess RFI severity.

the RFI measurements. From these measurements the power control room, containing all of the high-voltage switching equipment, had some of the highest emission levels in the building. As a result, the power control room will be used as an example case for our propagation model case study.

The emissions were characterised by measuring the field strength at 3 m away and calculating the EIRP. The shielding to the receiver is then derived from the steps contained in Sec. 6.5 for the case where the RFI source is contained in the KAPB. Fig. 6.20 shows the calculated levels at the victim receivers M59 and M60. To assess the severity of the RFI emission, the current SARAS levels for a 100 kHz resolution bandwidth is added for comparison. Both the two-ray and Egli ground loss models are used for the propagation calculation resulting in the respective upper and lower bound illustrated in Fig. 6.20. Receiver M59 and M60 do not benefit from the additional Losberg shielding, making these receivers a good worst-case study.

From these comparisons, it is clear that when the Egli model is used, the RFI level at the receivers is below the SARAS level. However, this does not entirely hold when using the two-ray model that gives a lower bound for the ground propagation. Some frequencies have the potential to exceed the SARAS levels. Nevertheless, meeting these stringent levels was only possible through the implementation of a number of excellent on-site EMC policies. However, in the case of a policy change, extra mitigation will need to be implemented to lower the emissions at the victim receivers M59 and M60.

6.8 Conclusion

In this part of the work, simple empirically-based propagation models were obtained using on-site measurements. This included characterising the shielding effectiveness of the KAPB using the Multi-copter, which significantly expedited measurement time over conventional methods. Also, existing ground-loss models were tested at varied heights over flat-earth. This was followed by large-scale diffraction measurements of the berm. Diffraction measurements were used to calibrate a single knife-edge diffractor. This data was also used to verify the computational model developed earlier in the dissertation. Combined, these models can now be used to quantify the effect of on-site noise sources without resorting to complex computational modelling. The model was also verified using a set of independent on-site measurements at discrete locations. Using the verified combined models, the effect of measured on-site emissions on the nearest MeerKAT receivers were investigated using the SARAS levels. Shielding towards these locations was found to be sufficient. However, increasing the stringency of the SARAS levels could require additional policies to be implemented.

Chapter 7

Conclusions and Recommendations

The investigation into propagation models for the Karoo Array Telescope has helped create an understanding of the site shielding characteristics. It is in the interest of the RFI environment to be able to predict the severity of on-site RFI sources. Quantifying the effect of these sources allows informed decisions about investing in additional shielding measures to be made. This work focussed on the characterisation of a man-made soil berm constructed next to the KAPB building. The structure was strategically placed to increase propagation loss to the nearest MeerKAT receivers. However, diffraction over this structure has brought its shielding performance into question. The objective of this work is summarised as follows:

1. Develop a practical computational model of the berm using scale-model and full-scale measurements.
2. Develop a Multi-copter measurement platform for full-scale propagation measurements and KAPB shielding investigations.
3. Use full-scale measurements to compile a set of empirical propagation models. These models can then be used for characterising the severity of RFI sources.

The size of the berm together with the frequencies of interest made practical computational modelling expensive. In the end, a model was conceived using a hybrid MoM and GO method. The method coupled with a PEC flat-earth approximation made shielding simulations possible. The latter approximation limited observations to non-LOS areas because of strong ground reflections. With full-scale measurements, this model was verified using the Multi-copter platform. It was then possible to simulate the shielding effectiveness of the berm for MeerKAT receiver M60 (5 km from the KAPB). These simulations

once again strengthened the finding that the berm has a shielding effectiveness of around 10 - 20 dB between 260 MHz and 930 MHz. It was also found that diffraction over the berm will never cause negative shielding when both the transmitter and receiver are located below its apex. Through further investigation with the model, it was found that the edges of the assembly shed structure, as well as other on-site metallic structures, could cause diffraction around the berm. These diffractions will lower the shielding effectiveness of the berm in some areas below 10 dB. To mitigate the loss in shielding effectiveness, it was proposed to minimise or to cover these metallic structures with a scattering or absorbing material. In the case of vehicles, it was recommended to shift their parking location to the far side of the assembly shed or at least to a point shielded by the berm. Other metallic structures to keep in mind are the water tower and overhead roof for the parking.

To characterise the berm structure, the shielding effectiveness of the KAPB and the ground loss, it was decided to develop a Multi-copter measurement vehicle. This vehicle significantly decreased conventional measurement time over large distances and made it possible to determine the diffraction over the berm up to a height of 100 m. During its development, key problems were identified such as characterising the on-board antenna pattern and self-generated RFI. A dual-broadband antenna scheme was implemented creating a quasi-isotropic pattern. To simplify modelling of the on-board antenna patterns, a shielded enclosure was built around all of the wiring and electronic systems. This enclosure also ensured pattern stability through multiple measurements and battery changes. The use of 3D-printable parts, general hardware parts and COTS flight electronics makes it possible to replicate the platform in other university groups.

Using the aerial-measurement platform, it was possible to characterise KAPB shielding at a height of 10 m. This height corresponded to a position in line with the centre of the building and the top edge of the berm. Shielding in this direction was on average in the order of 40 dB (260 MHz to 930 MHz). The berm itself was characterised using a simple single knife-edge diffractor. The height of this knife edge was chosen so that its predictions closely followed the vertical-diffraction measurements. The KAPB shielding together with the knife-edge model for the berm, and an empirically tested ground-loss (Egli Model) equation, made it possible to create a set of rules to calculate a shielding budget to nearby receivers. These empirically-verified models were selected to be simple enough to increase accessibility. Using a set of independent measurements, the models were tested at discrete locations. These models will make it possible for an RFI engineer to assess the severity of the noise source without resorting to complex computational modelling. These assessments should make it possible to make a more informed decision on passing or failing site equipment. Such decisions in themselves could have financial and time investment consequences. As an example, it was calculated that the maximum transmit power using SARAS of a source in the KAPB at

200 MHz and a bandwidth of 12.5 kHz would be about -42 dBm. Additionally, the effect of on-site emissions from the power control room to the nearest MeerKAT locations were evaluated against the SARAS levels. These levels were found to be acceptable. However, an increase in stringency will require shielding policies to be upgraded. Using the computational model developed in this work, new models could be created for other berm geometries. This becomes more relevant as on-site measurement campaigns become logistically harder as MeerKAT starts to come into operation.

Further investigations using the tested computational model are also possible such as simulating the actual shielding of Losberg. It is also possible to use this model to optimise the berm geometry in the case where more construction work is planned. Future work possible in terms of the Multi-copter includes replacing the resistively loaded antennas with more-efficient blade antennas. Also, fully integrating the time-domain receiver into the vehicle for on-site measurements.

This work has focussed on the characterising propagation of on-site RFI computational models and full-scale measurements. The result has been a deeper insight into berm diffraction, a verified computational model, a set of empirical propagation models and a metrology vehicle. These models, as well as the measurement vehicle, will increase confidence in assessing on-site RFI culprits.

Appendices

Appendix A

Waveguide Measurements

A.1 Calibration standards

These are the calibration standards used for the S-Band and X-Band waveguide measurement systems. (Tab. A.1, A.2, A.3 and A.4). Data was acquired directly from the Cal-Kit Manager software. This software keeps track of calibration information and can upload a calibration configurations to VNA memory slots.

Standard	Length (ps)	Start [GHz]	Stop [GHz]	Designation
Offset short	172.85	2.078	4.156	1
Offset short	289.17	2.078	4.156	2
Sliding Load	-	2.078	4.156	3
Thru	-	2.078	4.156	4

Table A.1: Calibration Standards defined for S-Band Waveguide system in CalKit Manager. Using a short, offset short, thru and sliding load calibration.

S11A	1
S11B	2
S11C	3
S22A	1
S22B	2
S22C	3
Forward Transmission	4
Reverse Transmission	4
Forward Match	4
Reverse Match	4

Table A.2: Calibration class assignments defined for S-Band Waveguide system in CalKit Manager.

Standard	Length (ps)	Start [GHz]	Stop [GHz]	Designation
Thru	0	6.5572	13.114	1
Thru	33.33	6.5572	13.114	2
Thru	133.333	6.5572	13.114	3
Short	0	6.5572	13.114	4

Table A.3: Calibration Standards defined for X-Band Waveguide system in CalKit Manager. Using a short, line and thru calibration.

TRL Thru	1
TRL Reflect	4
TRL Line or Match	2,3

Table A.4: Calibration class assignments defined for X-Band Waveguide system in CalKit Manager.

A.2 Calibration and repeatability validation

Here a multitude of measurements made on different occasions were compared for the sake of testing repeatability and gaining confidence in the measurements. Fig. A.1 and A.2 shows all of the S-Band and X-Band measurements in superposition. This gives a clear indication of the repeatability of the measurement and calibration.

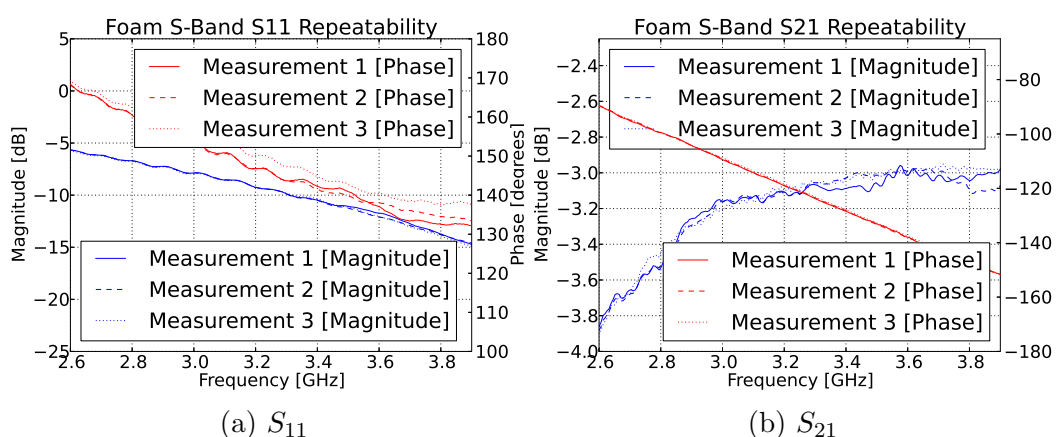


Figure A.1: S-Band measurements made over multiple occasions to illustrate repeatability of measurements.

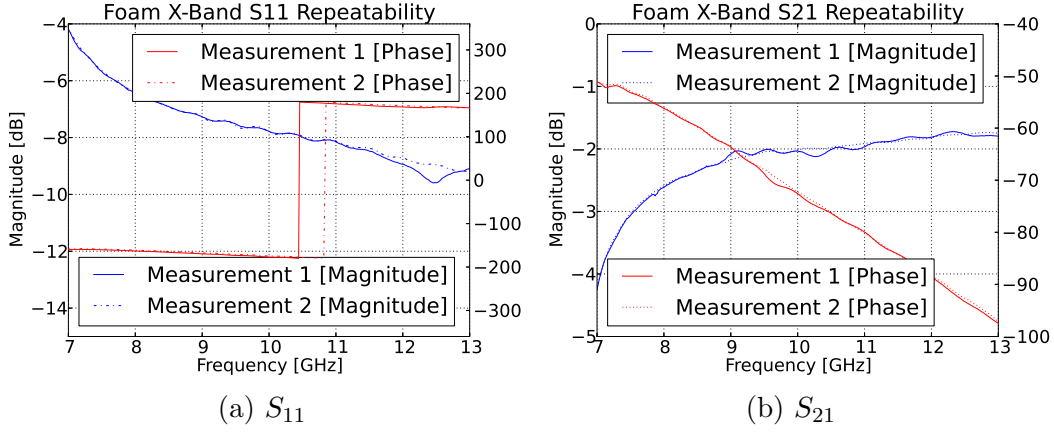


Figure A.2: X-Band measurements made over multiple occasions to illustrate repeatability of measurements.

A.3 Improved stepwise Nicolson-Ross-Weir Extraction

The extraction code used to find the permittivity of the dielectrics from their S-Parameters is shown here. The code in Listing A.1 uses the algorithm featured in [60]. As an illustration, Fig. A.3 show a result calculated with the typical and improved algorithm. The typical algorithm kept the fundamental root through the entire extraction and therefore deviates from its actual value. This is especially prone to happen with long sample lengths and materials with high permittivity.

The NRW method can be briefly demonstrated by its equations [60]. In these equations, the measured S-parameters are used to calculate the reflection

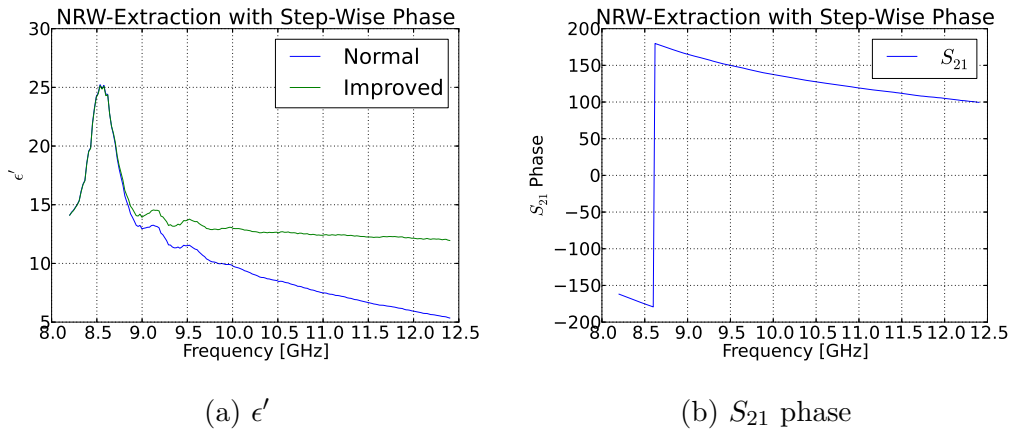


Figure A.3: Comparison between typical and improved NRW code. As a result of choosing the incorrect root, the extraction deviates. The point at which this happens is illustrated as the wrapping point in the phase graph.

coefficient from a semi-infinite slab (Eq. A.3.1).

$$\begin{aligned}\Gamma &= \chi \pm \sqrt{\chi^2 - 1} \\ \chi &= \frac{S_{11}^2 - S_{21}^2 + 1}{2S_{11}}\end{aligned}\quad (\text{A.3.1})$$

This result is then used to calculate the phase factor of the wave propagating through the material slab (Eq. A.3.2).

$$e^{-\gamma d} = \frac{S_{11} + S_{21} - \Gamma}{1 - (S_{11} + S_{21})\Gamma} \quad (\text{A.3.2})$$

where

- d is the thickness of the sample

Finally, the refractive index and permeability is calculated. In turn making it possible to also extract the permittivity (Eq. A.3.3).

$$\begin{aligned}n^2 = e_r \mu_r &= -\left[\frac{c}{\omega d} \ln(e^{\gamma d})\right]^2 + \left(\frac{\omega_c}{\omega}\right)^2 \\ \mu_r &= \left(\frac{1 + \Gamma}{1 - \Gamma}\right) \sqrt{\frac{n^2 - \left(\frac{\omega_c}{\omega}\right)^2}{1 - \left(\frac{\omega_c}{\omega}\right)^2}}\end{aligned}\quad (\text{A.3.3})$$

where

- e_r and μ_r is the relative permittivity and permeability
- c is the speed of light
- ω_c is the characteristic frequency
- ω is the frequency in question

The problem with the classical form of the NRW extraction can be seen in Eq. A.3.2. The periodicity of the wave propagating through the slab can have an infinite amount of roots. This is illustrated in Eq. A.3.4 where m represents some integer. In the classical extraction, m is always taken as 0. However, with longer sample lengths and dielectric parameters this assumption may become invalid.

$$e^{-\gamma d} = e^{-\alpha d} e^{-j\beta d + 2\pi m} \quad (\text{A.3.4})$$

With the stepwise algorithm, the phase is unwrapped through the whole frequency range. This unwrapping causes the m to be automatically set to the correct root.

Listing A.1: Python code used to extract ϵ_r and μ_r by using the NRW method with an unwrapped phase technique.

```

1      #Define free-space constants
2      eps0 = 8.854187817e-12
3      mu0 = 4*np.pi*1e-7
4      c0 = 1/np.sqrt(eps0*mu0)
5
6      #Calculate wave-guide cutoff frequency
7      fc = 1/(2*guideWidth*np.sqrt(mu0*eps0));
8
9      #Convert to angular frequency
10     wc = 2*np.pi*fc;
11     w = 2*np.pi*freq;
12
13     #Calculate correct gamma value
14     X = (np.square(S11) - np.square(S21) + 1)/(2*S11)
15     gamma = []
16     for XIter in X:
17         g = XIter + np.sqrt(np.square(XIter) - 1)
18         if np.abs(g) > 1:
19             gamma.append(XIter - np.sqrt(np.square(XIter) - 1))
20         else:
21             gamma.append(g)
22     Gamma = np.array(gamma)
23
24
25     z = np.sqrt((np.square(1 + S11) - np.square(S21))/(np.square
26         (1-S11) - np.square(S21)));
27     egamd = (1-np.square(S11)+np.square(S21))/(2*np.array(S21)) +
28         (2*np.array(S11))/((z - 1/z)*S21);
29
30     #Calculate unwrapped phase
31     phi0 = float(np.angle(egamd[0],0))
32
33     phi = [phi0]
34     tmp_phi = 0
35     for n in range(1,len(freq)):
36         for a in range(1,n):
37             tmp_phi = tmp_phi + float(np.angle(egamd[a]/egamd[a
38                 -1],0))
39             phi.append(phi0+tmp_phi)
40             tmp_phi=0
41     phi = np.array(phi)
42
43     #Calculate log of egamd with correct imaginary root
44     lnegamd = np.log(abs(egamd)) + 1j*phi
45
46     #Calculate propagation term and refractive index respectfully
47     nsqr = -1.*np.square((c0/(w*sampleLength))*(lnegamd)) + np.
48         square(wc/w)

```

```
47 | #Calculate relative dielectric parameters mu and eps  
48 | mur = ((1+Gamma)/(1-Gamma))*np.sqrt((nsqr - np.square(wc/w))  
49 |      /(1 - np.square(wc/w)))  
50 | epsr = nsqr/mur  
51 | return [epsr ,mur]
```

Appendix B

Free-space Dielectric Measurements

B.1 Fields in the focal region of horn antennas

Here, in Fig. B.1, the fields are shown directly out of [63]. The distribution in the focal region remains within 2 dB for at least 120 mm centred around the centre. This will be deemed acceptable for the measurements on the 115 mm wide dielectric block. The E-plane in this plot is vertically polarised with the z-axis positive away from the antenna and centred on the focal plane.

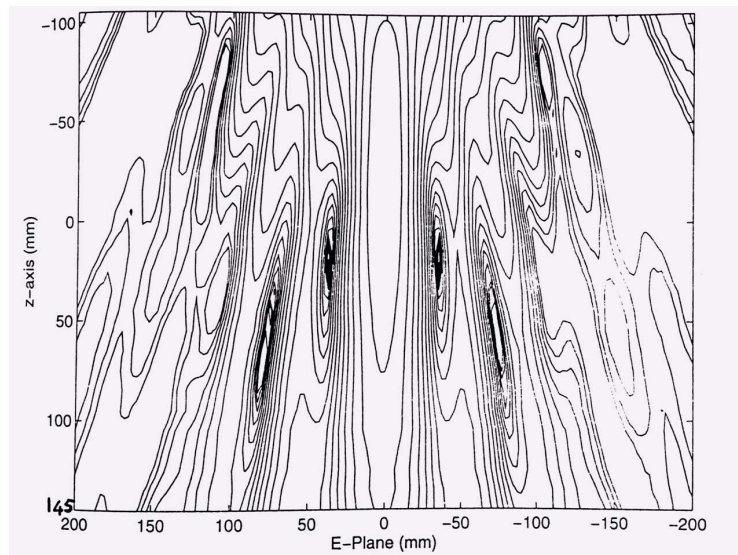


Figure B.1: Amplitude in dB of the radiation pattern in the E-plane along the z-axis for a frequency of 13.5 GHz. Difference between contours are 2dB with a -2dB contour in the middle.

B.2 Offline calibration of measured S-Parameters

A load, short and offset short offline calibration was done for the free-space system. The measurements were accomplished using a vector network analyser (VNA) calibrated up to the waveguide interface. The uncalibrated S-parameters that were used to calculate the error coefficients can be seen in Fig. B.2. These S-parameters include the effect of the Ku waveguide coupling as well as the horn antenna and lens interface. All of the effects are removed during calibration.

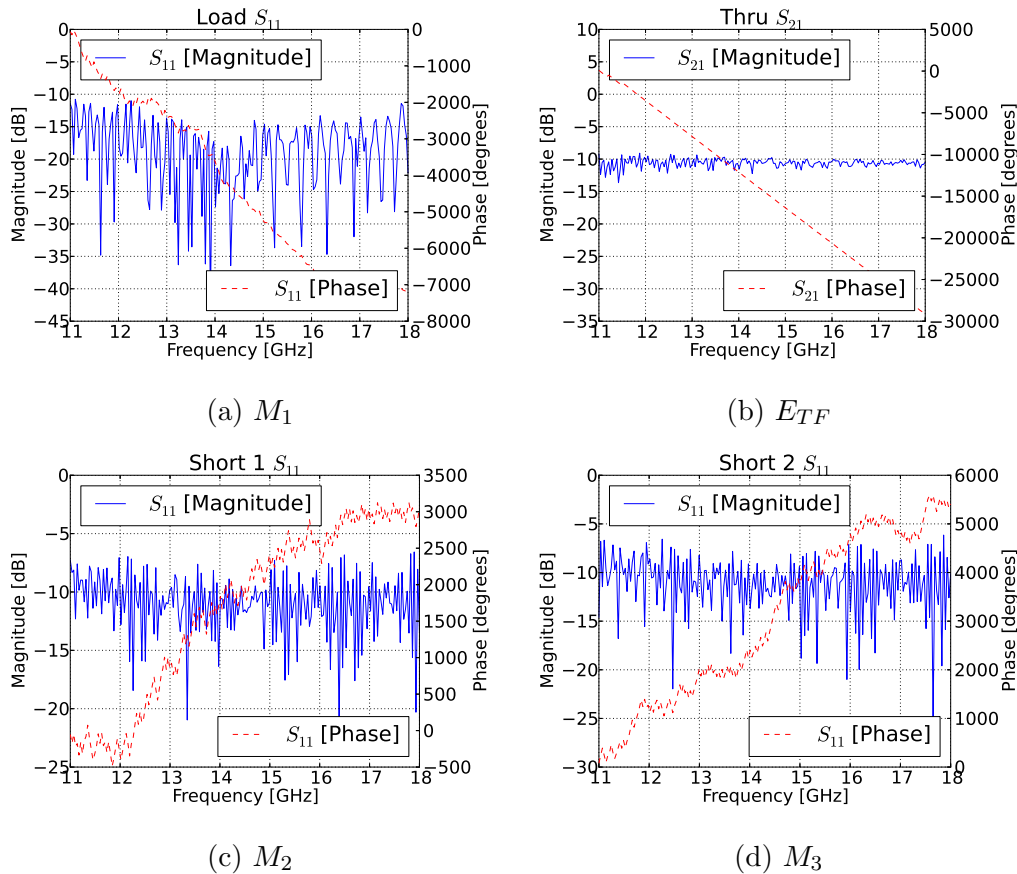
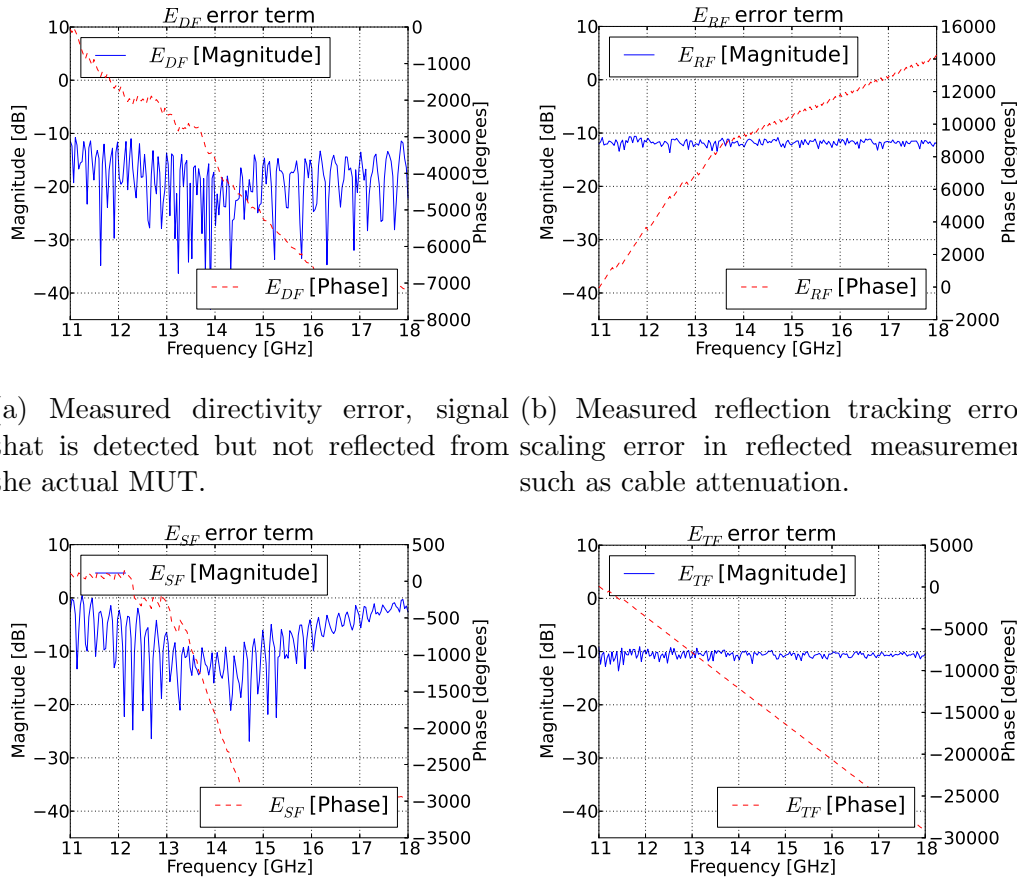


Figure B.2: Measured standards used in offline calibration.

B.3 Calculated error coefficients

The error coefficients used to do the offline calibration was calculated using the equations in Table 3.4 from the measured data in Fig. B.2. The measured error coefficients, as expected, are in very close agreement to the original dissertation linked to the system referenced earlier in this chapter. The actual error terms can be seen in Fig. B.3. These error terms are all that is needed to complete the offline calibration. For practical reasons E_{LF} was taken to be 0.



(a) Measured directivity error, signal (b) Measured reflection tracking error, that is detected but not reflected from scaling error in reflected measurement the actual MUT. such as cable attenuation.

(c) Measured source match error, the (d) Measured transmission frequency re-measured impedance mismatch in the sponse error, similar to reflection tracking system. but for the transmission path.

Figure B.3: Calculated error terms used to calibrate measurements offline.

B.4 Offline calibration and extraction code

During offline calibration, the measured standards are used to generate error coefficients discussed in Section B.3. These error coefficients are then used to extract the actual material parameters. The code used for the extraction can be seen in Listing. B.1.

Listing B.1: Python code used to do offline calibration for free-space extraction

```

1  import numpy as np
2
3  #Initial Parameters
4  #Define free-space constants
5  eps0 = 8.854187817e-12
6  mu0 = 4*np.pi*1e-7
7  c0 = 1/np.sqrt(eps0*mu0)
8
9  d = 0.115
10 root = 4
11
12 #Import S-Parameters
13 print("Importing _Calibration _S-Parameters")
14
15 #Import 1 Load, 2 Shorts and a Thru measurement
16 freq, S11_L2S_Load = fileIO.readSParamFromFile("
    dielectricFreespace_Cal_Load",True)
17 freq, S11_L2S_Short1 = fileIO.readSParamFromFile("
    dielectricFreespace_Cal_Short",True)
18 freq, S11_L2S_Short2 = fileIO.readSParamFromFile("
    dielectricFreespace_Cal_OffsetShort",True)
19
20 #Calculate frequency and wavelength based on the imported
    measurements
21 lamda = (c0*1.045)/freq
22 B = 2*np.pi/lamda
23
24 #Calculate error coefficients
25 M1 = S11_L2S_Load
26 M2 = S11_L2S_Short1
27 M3 = S11_L2S_Short2
28
29 S1 = 0
30 S2 = -1*np.exp(0)
31 S3 = -1*np.exp(1j*2*B*5.7e-3)
32
33 EDF = M1
34 ERF = ((S2-S3)*(M1-M3)*(M1-M2))/(S2*S3*(M3-M2))
35 ESF = (S3*(M2-M1) - S2*(M3-M1))/(S2*S3*(M3-M2))
36
37 ETF = S21_L2S_Thru;
38
39 #Set the measured data

```

```

40 freq , S11.L2S.Thru , S21.L2S.Thru , S12.L2S.Thru , S22.L2S.Thru =
    fileIO.readSParamFromFile("dielectricFreespace_Foam",False)
41 S11M = S11.L2S.Thru
42 S21M = S21.L2S.Thru
43
44 #Calculate actual S parameters
45 S11Foam = (S11M - EDF)/(ESF*(S11M-EDF) + ERF)
46 S21Foam = (S21M*(1-ESF*S11Foam))/(ETF)
47
48 #Shift phase by sample size
49 phase = (-(1j*2*np.pi*freq)/(c0*1.045))*d
50 S11FoamShifted = S11Foam*np.exp(2*phase)
51 S21FoamShifted = S21Foam*np.exp(phase)
52
53 #Extract the material parameters for the given S-parameters
54 epsr , mur = dea.nicolsonRossWeirWithStepWisePhaseTEM(freq ,
    S11FoamShifted , S21FoamShifted , d)
55
56 return freq , epsr , mur

```

B.5 Root combinations

To find the correct root in the free-space extraction a method from [65] was used. Here Eq. 3.3.5 was used, along with its uniqueness criterion where $\frac{P_1}{P_2}$ should be rational. Using the phase from the S-parameter measurements in Fig. 3.9, root combinations were found with the least amount of error seen in Fig. B.4. The minimum error for each of these combinations can be seen plotted in Fig. B.5, with a red line marking the three lowest error combinations.

The combinations with the least error can be seen tabulated in Table 3.5 with their respective 2D plots seen in Fig. 3.10, B.6 and B.7. The code used to calculate the combination for least error can be seen in Listing B.2.

Listing B.2: Python code used to calculate the combination with the least error

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 #Calculate some general parameters
5 gam = 1j*((2*np.pi)/(c0/11e9))*(np.sqrt(2.1))
6 invDelta = 1j*(gam/(2*np.pi))
7 sampleWaveLength = 1/invDelta
8 ks = (2*np.pi)/sampleWaveLength
9 ks = np.real(-1*ks)
10
11 #Initialise parameters being used by the search
12 smallestError = 100
13 bestP1 = 0
14 bestP2 = 0

```

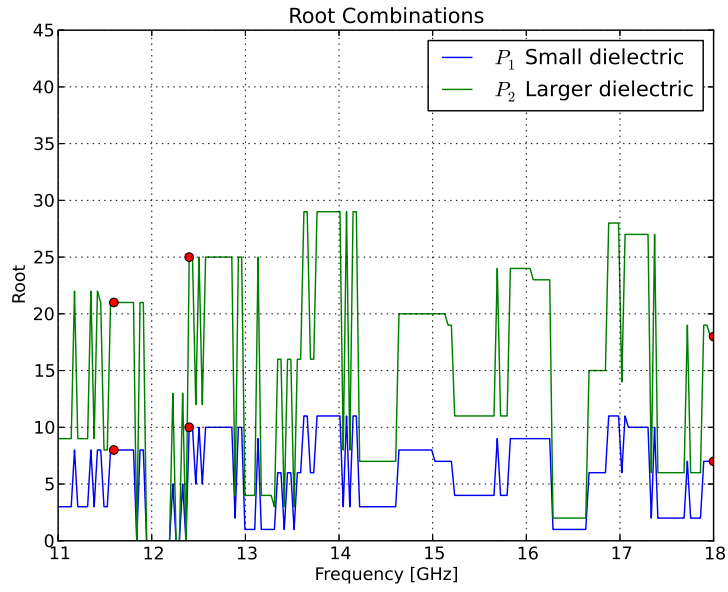


Figure B.4: Root combinations found over frequency for minimum error using Eq. 3.3.5.

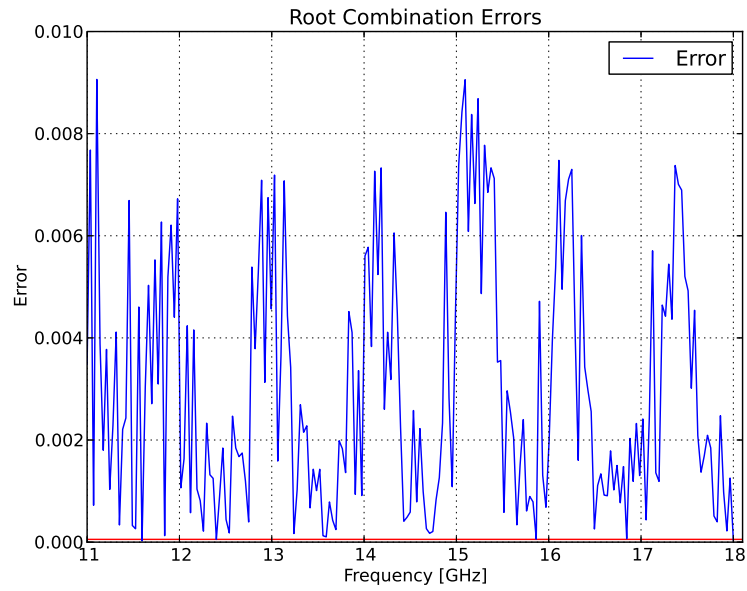


Figure B.5: Root combinations error over frequency.

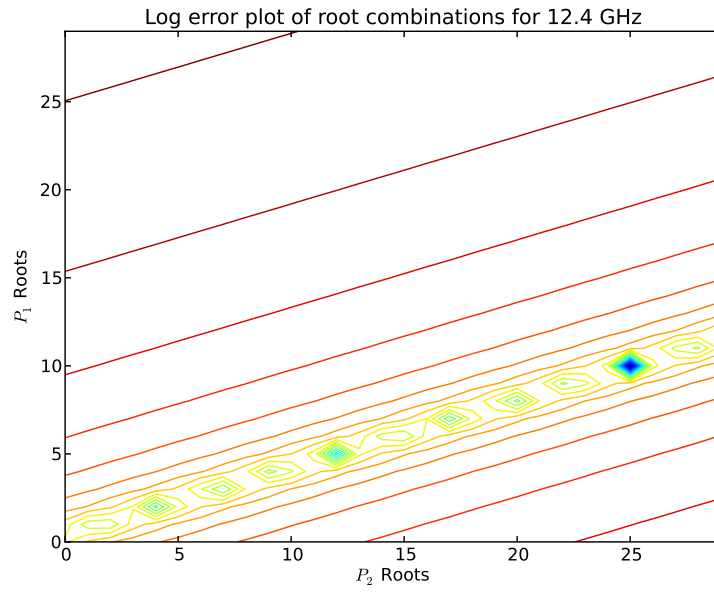


Figure B.6: Root combination minimum error plot at 12.39 GHz. This plot solves the left-hand side of Eq. 3.3.5.

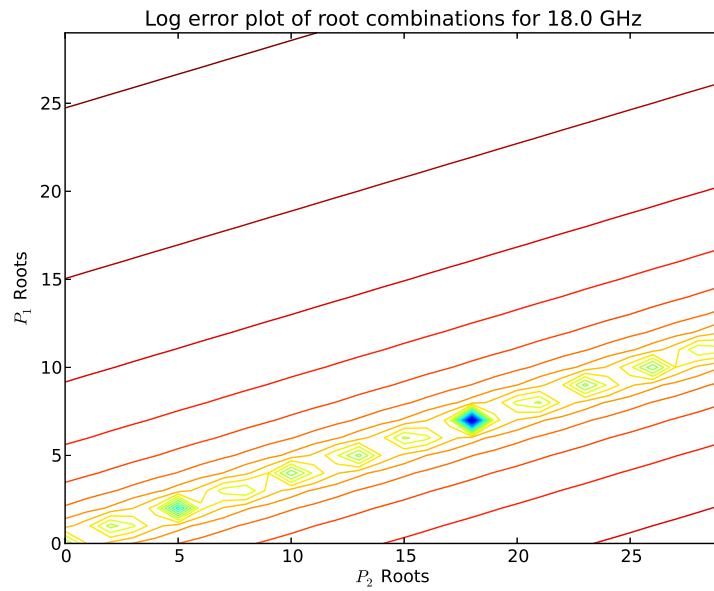


Figure B.7: Root combination minimum error plot at 18 GHz. This plot solves the left-hand side of Eq. 3.3.5.

```

15
16 #Initialise storage parameters
17 p1g = []
18 p2g = []
19 errorg = []
20 p1Range = np.arange(0,30)
21 p2Range = np.arange(0,30)
22 errorMap = np.zeros(30*30).reshape((30, 30))
23
24 #Define the length of the 2 materials
25 d1 = 0.115
26 d2 = 0.3
27
28 #Iterate through the set range of roots for each frequency
29 for i in np.arange(0,len(freq)):
30     for p1 in p1Range:
31         for p2 in p2Range:
32
33             #Get the phase from the measured S-Parameter
34             argZ1 = np.angle(S21FoamCShifted[i])
35             argZ2 = np.angle(S21FoamCBigShifted[i])
36
37             #Calculate the left and right-hand side of the
38             equation
39             left = (d2*argZ1 - d1*argZ2)/(2*np.pi)
40             right = p2*d1 - p1*d2
41
42             #Calculate the error term for this combination of
43             roots
44             error = np.abs(left-right)
45
46             #Add this error to the 2D plot
47             errorMap[p1][p2] = 20*np.log(error)
48
49             #Check and update the best root combination
50             if(error < smallestError):
51                 smallestError = error
52                 print("P1: "+str(p1)+" P2: "+str(p2)+" Error: "+
53                     str(error))
54                 bestP1 = p1
55                 bestP2 = p2
56
57             #Store the best root combination for this frequency
58             p1g.append(bestP1)
59             p2g.append(bestP2)
60             errorg.append(smallestError)
61
62             #Draw a 2D visualisation of the current root combinations
63
64             X,Y = np.meshgrid(p1Range, p2Range)
65             plt.figure()
66             plt.title("Log_error_plot_of_root_combinations_for "+str(freq[

```

```

        i]*1e-9)+"_GHz")
64     plt.xlabel("$P_2$ Roots")
65     plt.ylabel("$P_1$ Roots")
66     plt.contour(X,Y,errorMap,25)
67     plt.xlim(0,29)
68     plt.ylim(0,29)
69     plt.show()
70
71 #Plot the best root combinations over frequency
72     plt.figure()
73     plt.title("Root_Combinations")
74     plt.xlabel("Frequency_[GHz]")
75     plt.ylabel("Root")
76     plt.plot(freq*1e-9,p1g,label="$P_1$ Small dielectric")
77     plt.plot(freq*1e-9,p2g,label="$P_2$ Larger dielectric")
78     plt.ylim(0,45)
79     plt.grid()
80     plt.legend()
81
82     plt.figure()
83     plt.title("Root_Combination_Errors")
84     plt.xlabel("Frequency_[GHz]")
85     plt.ylabel("Error")
86     plt.axhline(lowestError,color="red")
87     plt.plot(freq*1e-9,errorg,label="Error")
88     plt.xlim(11,18.1)
89     plt.grid()
90     plt.legend()
91     plt.show()

```

B.6 Parameter fitting

To compress the extracted dielectric information, analytical curve fitting is used. These curves are calculated from the measurements using the code in Listing B.3. The code starts by combining the three dielectric measurements into a single dataset. This dataset is then divided into a test and fitting set. The test set is used to determine the quality of the resulting fit. This quality factor is also used as a metric to determine the order of the polynomial fit. The real and imaginary parts are handled individually. The real part is used to determine the order of the polynomial.

Listing B.3: Python code used to fit extracted dielectric data to a polynomial and exponential curve.

```

1 import numpy as np
2 from scipy.optimize import curve_fit
3
4 def curveFit()
5     #Build up a single data set with all measurements
6     freq1,epsr1,mur1 = SBandExtractionFoam()

```



```

7     freq2, epsr2, mur2 = XBandExtractionFoam()
8     freq3, epsr3 = FreespaceExtractionFoam()
9     freq = np.concatenate((freq1, freq2, freq3))
10    epsr = np.concatenate((epsr1, epsr2, epsr3))
11
12    #Choose a random 80 percent of fitting points and
13    #store the rest as testing points
14    freqFit = freq
15    epsrFit = epsr
16
17    freqTest = []
18    epsrTest = []
19
20    size = len(freq)
21    while len(freqTest)/size < 0.2:
22        #Generate random number
23        randIndex = rnd.randint(0, len(freqFit)-1)
24        #Shift this data point to the test set
25        freqTest.append(freqFit[randIndex])
26        epsrTest.append(epsrFit[randIndex])
27
28        freqFit = np.delete(freqFit, randIndex)
29        epsrFit = np.delete(epsrFit, randIndex)
30
31    freqTest = np.array(freqTest)
32    epsrTest = np.array(epsrTest)
33
34    #Fit Polynomial Curve
35    #Calculate the least error fit using the test data for
36    #a polynomial curve. The real and imaginary curves are
37    #handled individually, using the order determined with
38    #the real data.
39    bestOrder = 0
40    bestScore = 0
41    for order in np.arange(0, 15):
42        coefPolyReal = np.polyfit(freqFit, np.real(epsrFit), order)
43
44        error = 0
45        for testIndex in np.arange(0, len(freqTest)):
46            linePoint = 0
47            i = len(coef) - 1
48
49            for c in coef:
50                linePoint = linePoint + c*((freqTest[testIndex]
51                                                    ))**i)
52
53                i -= 1;
54
55            error = error + np.abs(np.real(epsrTest[testIndex]) -
56                                    linePoint)

```

```
57         bestScore = error
58     else:
59         if(bestScore > error ):
60             bestScore = error
61             bestOrder = order
62
63     coefPolyImag = np.polyfit(freqFit,np.imag(epsrFit),order)
64
65     #Fit Logarithmic Curve
66     coefLogReal, pcov = curve_fit(func, np.array(freqFit), np.
67         real(epsrFit))
67     coefLogImag, pcov = curve_fit(func, np.array(freqFit), np.
68         imag(epsrFit))
69
69     return coefPolyReal, coefPolyImag, coefLogReal, coefLogImag
70
71 def func(x, a, b):
72     return a * np.log(x) + b
```

Appendix C

Coaxial Measurements

C.1 Material parameter extraction

The coaxial probe measurements were made using two available systems [85] [54]. As discussed in Chapter 3 the measurements with the probes were not used in the final characterisation of the dielectric. This was decided because of the porous nature of the material. The two different probes were both developed in-house.

All of the probe measurements were made in an air-conditioned room. Additionally, measurements were completed in the shortest time-scale possible following calibration. To test repeatability, each material was measured three times. Different calibrations were used for the two systems. The shielded probe [54] used a short, open and load scheme whereas the probe from Marais [85] used an open, line and thru configuration. A reference measurement of air was made before measuring the materials that were used in each extraction. Fig. C.1 and Fig. C.2 shows the Russouw [54] and Marais [85] probes with their respective calibration standards. To ensure reliable measurements, it was first important to measure some known dielectrics. In this case well known, Rexolite and Teflon samples were used (see Fig. C.3). Afterwards, each measurement is extracted with a known reference, in this case air was used. The latter is, usually, chosen as close as possible to the extracted material to improve accuracy. Table C.1 shows the expected dielectric parameters compared to the measurements for each material. These material were well characterised and noted in [86]. Dielectric parameters were extracted using the Morris Rimbi code. This code can be found in Section C.2.

With these results, it can be seen that the Rossouw probe showed the closest agreement with known values. The Marais probe only deviated with the measurement of the Rexolite. Further measurement discussions will only focus on that of the Rossouw probe. The measurements of this probe can be seen in Fig. C.4. These measurements were made in the same calibration set as the known measurements in Fig. C.1.

In this measurement, it is clear that the measured value changes considerably from one measurement to the next. The same trend was observed using the Marais probe. This effect was in the end attributed to the structure of the foam. Because of the small probe size relative to the structure of the foam, each measurement is a function of the probe position in this structure. Therefore, accurate coaxial measurements are not possible using a probe of this size. It is possible to use a larger probe. However, the frequency range of such a probe will be irrelevant to this project. In the case of the waveguide and free-space measurements, the probing area is spread over a much larger area. This reduces and averages the effect seen with the coaxial probes.

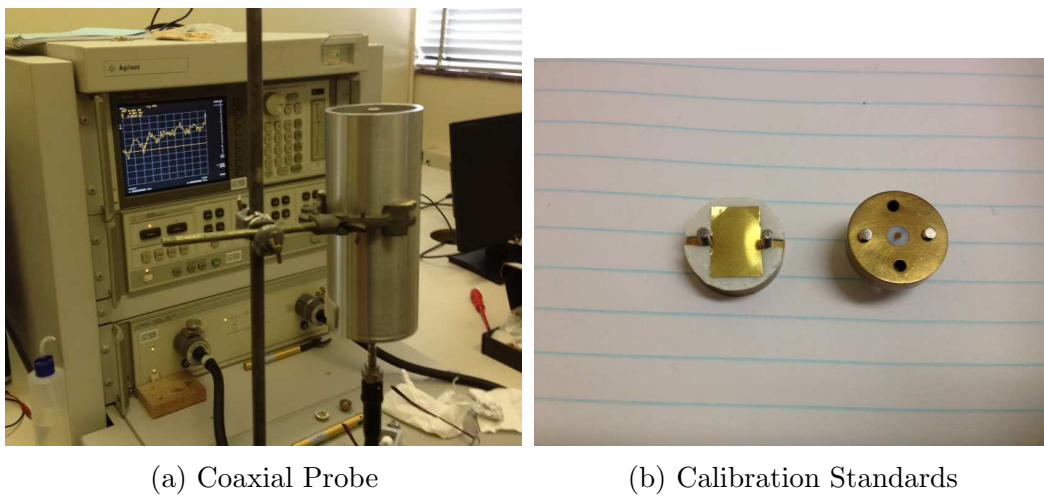


Figure C.1: Russouw Probe with its calibration standards. A short on the left and matched load on the right.

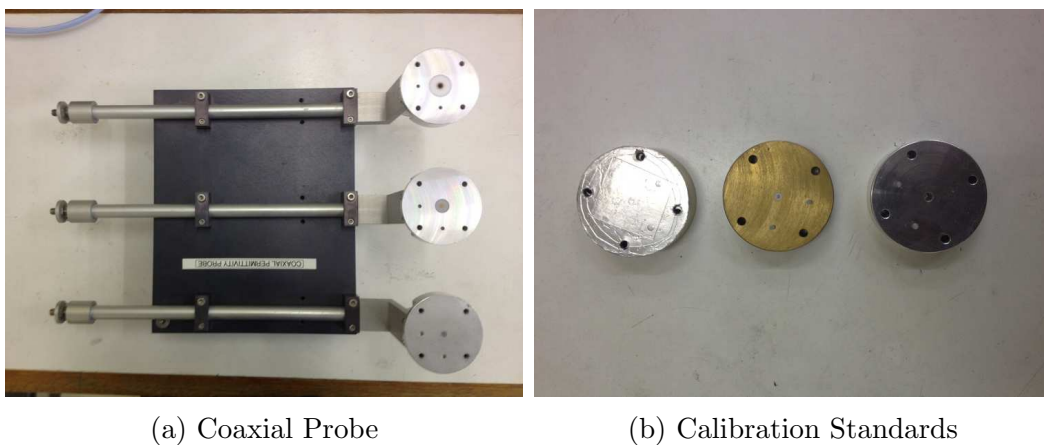


Figure C.2: Marais Probe with its calibration standards. From left; a short, matched load and a characterised open.



Figure C.3: Samples used in coaxial probe measurements. From left; Foam material, Rexolite, Perspex and Teflon.

	Marais	Rossouw	Known
Perspex	2.39, 2.39, 2.41	2.39, 2.34, 2.28	2.1 - 3.9
Rexolite	2.04, 1.91, 1.90	2.56, 2.57, 2.51	2.56
Teflon	2.01, 2.05, 1.69	1.98, 1.93, 1.92	2.065

Table C.1: Real permittivity of known materials, measured using coaxial probes. Each measurement is repeated three times, each with a new calibration.

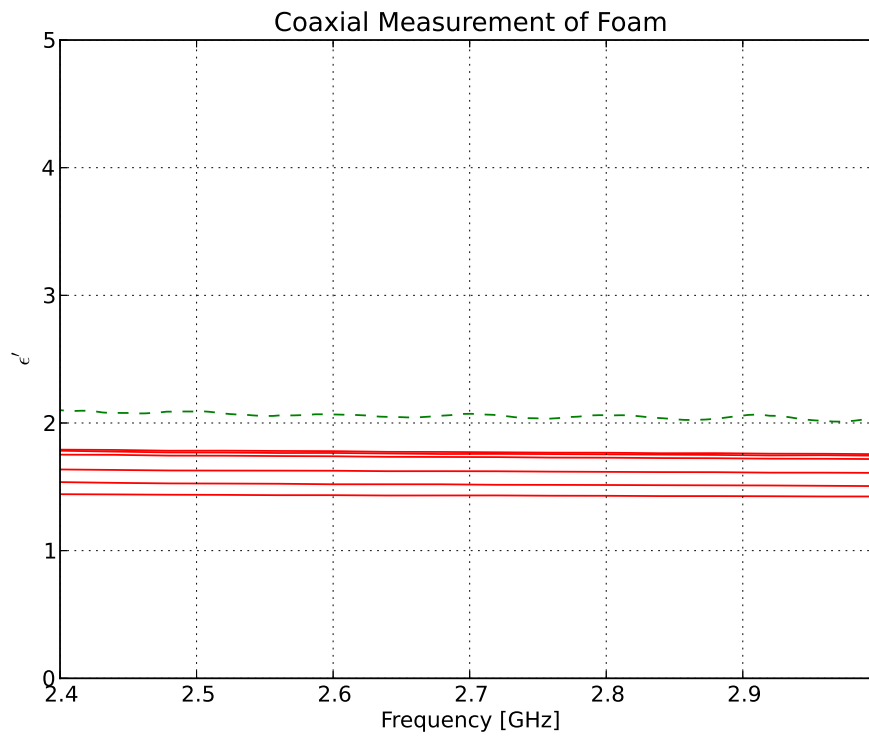


Figure C.4: Extracted foam measurements using Rossouw probe. The influence of the material structure can be seen in the random deviations of the measurements. The dashed line indicates the waveguide measurement for the same band.

C.2 Morris Rimbi code

The code used to extract measurements can be seen in Listing C.1. This code was translated from the original Matlab version found in the Rimbi thesis [57].

Listing C.1: Python code used for coaxial measurement extractions

```

1 import numpy as np
2
3 def morrisRimbi(freq, refS11, dataS11, innerConRadius, outerConRadius,
  refMaterialEr = 1.00064):
4     #Magnitude of known material
5     s = np.abs(refS11)
6     #Phase of known material
7     p = np.angle(refS11)
8     dphi = np.abs(p)
9     #Angle in degrees
10    phase = (p*180)/np.pi
11
12    #Magnitude of unknown material
13    s11 = np.abs(dataS11)
14    #Phase of unknown material
15    p11 = np.angle(dataS11)
16    #Angle in degrees
17    phase11 = (p11*180)/np.pi
18
19    #Frequency in radians
20    w = (2*np.pi)*freq
21    #Characteristic impedance
22    z = 50
23    #Radius of innerconductor
24    b = innerConRadius
25    #Radius of outerconductor
26    a = outerConRadius
27    #Speed of light
28    c = 2.99792458e8
29    #Wavelength in air
30    lamda = c/freq
31
32    #Reference material permittivity
33    er = refMaterialEr
34
35    #Parameters to be used for permittivity calculation
36    Co = np.tan(dphi/2)/(w*z)
37    #Capacitance as a result of fringing fields in the air
38    K = sp.ellipk(2*np.sqrt(a*b)/(a+b))
39    #Complete elliptical integral of the first kind
40    E = sp.ellipse(2*np.sqrt(a*b)/(a+b))
41    #Complete elliptical integral of the second kind
42    Ct = (8*(a+b)*(E - 1))/(z*w**lamda*np.log(a/b))
43    #Total capacitance of the probe
44    #Capacitance inside probe dielectricum
45    Cf = 0;

```

```
46  
47     #Calculate permittivity  
48     k = Cf/Co  
49     nume1 = (2*s11*np.sin(-p11))  
50     dene1 = (z*w*Co)  
51     dene12 = (1+2*s11*np.cos(p11) + s11**2)  
52     den = dene1*dene12  
53     nume2 = (1-s11**2)  
54  
55     e1 = (nume1/den)-k  
56     e11 = (nume2)/((z*w*Co)*(1+2*(s11**2)*np.cos(p11) + s11**2))  
57     epsr = e1 - 1j*e11  
58     return epsr
```


Appendix D

Quasi-infinite Plate Optimisation Program

The serration optimiser seeks the optimal edge geometry of a serrated plate with the goal of minimising the amount of diffraction onto itself. The serration geometry is fixed to a specified length and a minimum width. This means that only the apex of the triangle can be adjusted between the triangular base points of each serration. Consequently, this reduces the amount of optimisable variables to a single parameter for each serration.

The process starts with placing the first edge. Each edge is initially placed perpendicular to the plate. An optimiser then shifts the angle of this edge to

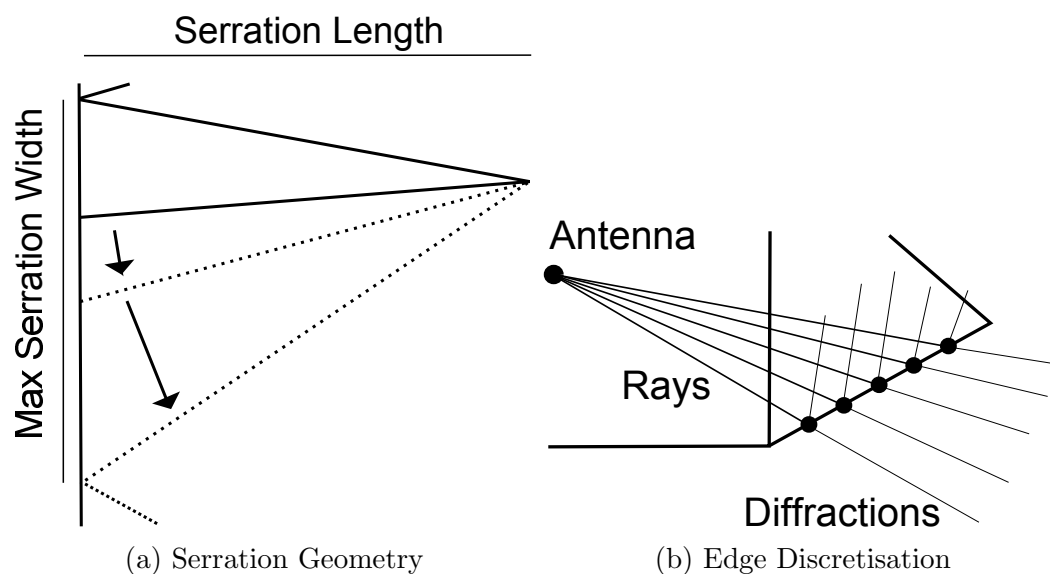


Figure D.1: In (a), an illustration of variable edges used to build the optimised geometry. The first edge here is already fixed while the second edge is being varied to optimise its angle. Additionally, the discretisation of an edge to calculate diffraction interactions can be seen in (b).

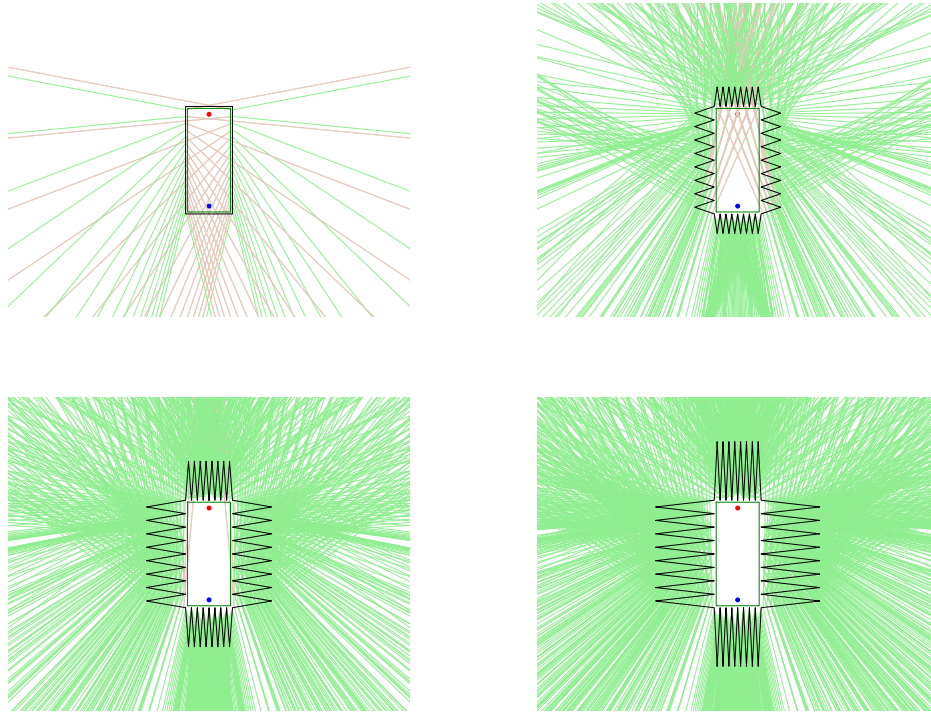


Figure D.2: Illustration of diffraction on a serrated plate. A simple non-optimised serration geometry was used, varying length of the vertexes.

minimize the number of rays travelling onto the plate (see Fig. D.1a). This is repeated for each edge until a symmetrical quarter of the plate is finished. This geometry is then duplicated over the symmetry axis. Individually, each edge is discretised into a set number of evenly spaced points (see Fig. D.1b). These points are used as the interaction points for rays transmitted by the source. Using the available geometry, the diffracted rays can be calculated. These diffracted rays are used to calculate an optimisation score, determined by the number of rays entering the specified clean zone.

To illustrate how the serrations minimise diffraction into the clear zone, a series of simulations is shown in Fig. D.2. A simple geometry was chosen and then varied for different lengths of serrations. This clearly demonstrates the advantage of using serrations.

The code used to optimise the plate geometry can be seen in Listing. D.1. The final coordinates for the optimised plate can be seen in Table D.1.

Listing D.1: Python code used to minimise diffractions onto a plate by optimising serrations

```

1 import numpy as np
2 from pylab import *
3
4 def getEdge(pointsX , pointsY , edgeNo) :
```

X (mm)	Y (mm)
687.5	0
1437.5	225
687.5	375
537.5	1125
312.5	375
112.5	1125
0	375

Table D.1: Coordinates for one-quarter of the optimised plate.

```

5   edgeX = np.array((pointsX[edgeNo], pointsX[edgeNo + 1]))
6   edgeY = np.array((pointsY[edgeNo], pointsY[edgeNo + 1]))
7   return edgeX, edgeY
8
9   def calcLineEquation(point1X, point1Y, point2X, point2Y):
10      m = (point2Y - point1Y)/(point2X - point1X) #Gradient
11      c = point2Y - m*point2X
12      return m, c
13
14   def getDiscPoints(edgeX, edgeY, discSize):
15      #Calculate discrete points on edge by first calculating the line equation
16      and filling an array with all of the discrete points
17      m, c = calcLineEquation(edgeX[0], edgeY[0], edgeX[1], edgeY[1])
18      length = np.sqrt((edgeX[1] - edgeX[0])**2 + (edgeY[1] - edgeY[0])**2)
19      #Number of discrete points
20      n = np.floor(length/discSize)
21
22      #Calculate midpoint
23      midX = edgeX[0] + (edgeX[1] - edgeX[0])/2
24      midY = edgeY[0] + (edgeY[1] - edgeY[0])/2
25      #Calculate segment length
26      segmentX = (np.abs(edgeX[1] - edgeX[0])/n)
27
28      #Calculate the placement offset
29      if(n%2 == 1): #If no of points are uneven
30          offsetX = 0
31      else: #If n is even
32          offsetX = segmentX/2
33
34      discListX = []
35      discListY = []
36      #Populate discrete points array
37      for i in np.arange(0, np.ceil(n/2)):
38          if offsetX == 0: #add only 1 point
39              discListX.append(midX)
40              discListY.append(m*(discListX[-1]) + c)
41          else: #add 2 points
42              discListX.append(midX + offsetX)
43              discListY.append(m*(discListX[-1]) + c)
44              discListX.append(midX - offsetX)
45              discListY.append(m*(discListX[-1]) + c)
46          offsetX = offsetX + segmentX
47
48      discListX = np.array(discListX)
49      discListY = np.array(discListY)
50
51      return discListX, discListY
52
53   def getAngleBetweenLines(m1, m2):

```

```

53     angle = np.arctan((m1-m2)/(1+m1*m2))
54     return angle
55
56 def getMirrorPoint(mMirror, cMirror, pointX, pointY):
57     d = (pointX + (pointY-cMirror)*mMirror)/(1+mMirror**2)
58     mirrorPointX = 2*d - pointX
59     mirrorPointY = 2*d*mMirror - pointY + 2*cMirror
60     return mirrorPointX, mirrorPointY
61
62 def generatePlate(lenX, lenY, wedgeLength, noOfWedges):
63     wedgeSize = lenX/noOfWedges
64
65     #Generate 1 quarter of the plate
66     structX = []
67     structY = []
68
69     #Generate the top part
70     for wedgeNo in np.arange(0, noOfWedges/2):
71         structX.append(wedgeNo*wedgeSize)
72         structY.append(lenY/2)
73         structX.append((wedgeNo+0.5)*wedgeSize)
74         structY.append(wedgeLength + lenY/2)
75
76     structX.append((wedgeNo+1)*wedgeSize)
77     structY.append(lenY/2)
78     wedgeSize = lenY/noOfWedges
79
80     #Generate the lower part
81     for wedgeNo in np.arange(0, noOfWedges/2):
82         structY.append((noOfWedges/2-wedgeNo)*wedgeSize)
83         structX.append(lenX/2)
84         structY.append(((noOfWedges/2-wedgeNo)-0.5)*wedgeSize)
85         structX.append(wedgeLength + lenX/2)
86
87     structY.append(0)
88     structX.append(lenX/2)
89
90     #Mirror this section to the other 3 quadrants
91     for i in np.arange(0, noOfWedges*2+1):
92         structX.append(structX[-2 - i*2])
93         structY.append(-1*structY[-2 - i*2])
94     for i in np.arange(0, noOfWedges*4+2):
95         structX.append(-1*structX[-2 - i*2])
96         structY.append(structY[-2 - i*2])
97
98     structX = np.array(structX)
99     structY = np.array(structY)
100
101     return structX, structY
102
103 def checkIfLinesCross(line1X, line1Y, line2X, line2Y):
104     #Calculate line equations
105     m1, c1 = calcLineEquation(line1X[0], line1Y[0], line1X[1], line1Y[1])
106     m2, c2 = calcLineEquation(line2X[0], line2Y[0], line2X[1], line2Y[1])
107
108     #Check if lines are parallel
109     if(m1 == m2):
110         return False, 0, 0
111
112     #Calculate intersection point
113     yIntersection = (m1*c2 - c1*m2)/(m1-m2)
114     xIntersection = ((c2-c1)/(m1-m2))
115
116     #Check if intersection point is within line bounds
117     if(xIntersection > np.min(line1X) and xIntersection < np.max(line1X) and
        yIntersection > np.min(line1Y) and yIntersection < np.max(line1Y)):

```

APPENDIX D. QUASI-INFINITE PLATE OPTIMISATION PROGRAM 146

```

118         if(xIntersection > np.min(line2X) and xIntersection < np.max(line2X)
           and yIntersection > np.min(line2Y) and yIntersection < np.max(
           line2Y)):
119             return True,xIntersection,yIntersection
120
121     return False,0,0
122
123 #Checks if any line enters a defined rectangle, note that this algorithm is
only for finite lines and a clockwise defined rectangle
124 def checkIfLineEntersClearZone(lineX,lineY,zoneX,zoneY,plotting=False):
125     #first check if there is no potential for a crossing
126     if(lineX[0] < np.min(zoneX) and lineX[1] < np.min(zoneX) or lineX[0] > np.
       max(zoneX) and lineX[1] > np.max(zoneX) or lineY[0] < np.min(zoneY) and
       lineY[1] < np.min(zoneY) or lineY[0] > np.max(zoneY) and lineY[1] >
       np.max(zoneY)):
127         return False
128
129     m,c = calcLineEquation(lineX[0],lineY[0],lineX[1],lineY[1])
130     crossed = False
131     x = 0
132     y = 0
133
134     #Check top edge
135     intersectPointX = (zoneY[0] - c)/m
136     if(intersectPointX > np.min(zoneX) and intersectPointX < np.max(zoneX)):
137         crossed = True
138         x = intersectPointX
139         y = zoneY[0]
140
141     #Check bottom edge
142     intersectPointX = (zoneY[2] - c)/m
143     if(intersectPointX > np.min(zoneX) and intersectPointX < np.max(zoneX)):
144         crossed = True
145         x = intersectPointX
146         y = zoneY[2]
147
148     #Check right Edge
149     intersectPointY = m*zoneX[1] + c
150     if(intersectPointY > np.min(zoneY) and intersectPointY < np.max(zoneY)):
151         crossed = True
152         x = zoneX[1]
153         y = intersectPointY
154
155     #Check left Edge
156     intersectPointY = m*zoneX[0] + c
157     if(intersectPointY > np.min(zoneY) and intersectPointY < np.max(zoneY)):
158         crossed = True
159         x = zoneX[0]
160         y = intersectPointY
161
162     if(crossed):
163         if(plotting): plt.plot(lineX,lineY,color="pink")
164         return True
165     else:
166         return False
167
168
169
170 def findBadRays(srcX,srcY,structureX,structureY,clearZoneX,clearZoneY,screenX,
       screenY,discreteSize,plotting,showBadRays=False):
171
172     plotLength = screenX + screenY
173
174     badRays = 0
175     #Loop through each available edge
176     for edge in np.arange(0,len(structureX)-1):

```

```

177     edgeX, edgeY = getEdge(structureX, structureY, edge)
178
179     #Calculate discrete points on edge
180     discListX, discListY = getDiscPoints(edgeX, edgeY, discreteSize)
181
182     #Loop through discrete points on chosen edge
183     for point in np.arange(0, len(discListX)):
184         #Calculate the incident angle of line on edge at specific point
185         mInc, cInc = calcLineEquation(srcX, srcY, discListX[point], discListY[
            point])
186
187         #Calculated the 2 diffracted rays (1 is the same as the incident
            ray the other ray needs some trigonometry)
188         ray1m = mInc
189         ray1c = cInc
190
191         #Get mirrored point around edge
192         #Calculate a point on the incident ray line
193         rayPointX = plotLength
194         rayPointY = (ray1m*rayPointX) + ray1c
195         rayPointX2 = -plotLength
196         rayPointY2 = (ray1m*rayPointX2) + ray1c
197         #mirror this point around the edge
198         mEdge, cEdge = calcLineEquation(edgeX[0], edgeY[0], edgeX[1], edgeY
            [1])
199         rayMirrorPointX, rayMirrorPointY = getMirrorPoint(mEdge, cEdge,
            rayPointX, rayPointY)
200         rayMirrorPointX2, rayMirrorPointY2 = getMirrorPoint(mEdge, cEdge,
            rayPointX2, rayPointY2)
201
202         #calculate the equation of this new mirrored point going through
            the original discrete Point
203         ray2m, ray2c = calcLineEquation(rayMirrorPointX, rayMirrorPointY,
            discListX[point], discListY[point])
204
205         #Calculate the no of rays passing through the clean zone
206         if(discListY[point] >= srcY):
207             if(ray1m <= 0):
208                 if(discListX[point] >= srcX):
209                     if(checkIfLineEntersClearZone(
                        np.array((discListX[point], rayMirrorPointX)), np.
                        array((discListY[point], rayMirrorPointY)),
                        clearZoneX, clearZoneY,
                        showBadRays)):
210                         badRays += 1
211                 else:
212                     if(checkIfLineEntersClearZone(np.array((discListX[
                        point], rayMirrorPointX2)), np.array((discListY[
                        point], rayMirrorPointY2)), clearZoneX, clearZoneY,
                        showBadRays)):
213                         badRays += 1
214                 else:
215                     if(discListX[point] >= srcX):
216                         if(checkIfLineEntersClearZone(np.array((discListX[
                        point], rayMirrorPointX)), np.array((discListY[point],
                        rayMirrorPointY)), clearZoneX, clearZoneY,
                        showBadRays)):
217                             badRays += 1
218                     else:
219                         if(checkIfLineEntersClearZone(np.array((discListX[
                        point], rayMirrorPointX2)), np.array((discListY[point],
                        rayMirrorPointY2)), clearZoneX, clearZoneY,
                        showBadRays)):
220                             badRays += 1

```

```

221         else:
222             if (raylm >= 0):
223                 if (discListX [point] >= srcX):
224                     if (checkIfLineEntersClearZone(np.array((discListX [
                        point], rayMirrorPointX)), np.array((discListY [point
                        ], rayMirrorPointY)), clearZoneX, clearZoneY,
                        showBadRays)):
225                         badRays += 1
226                     else:
227                         if (checkIfLineEntersClearZone(np.array((discListX [
                        point], rayMirrorPointX2)), np.array((discListY [point
                        ], rayMirrorPointY2)), clearZoneX, clearZoneY,
                        showBadRays)):
228                             badRays += 1
229                 else:
230                     if (discListX [point] >= srcX):
231                         if (checkIfLineEntersClearZone(np.array((discListX [
                        point], rayMirrorPointX)), np.array((discListY [point
                        ], rayMirrorPointY)), clearZoneX, clearZoneY,
                        showBadRays)):
232                             badRays += 1
233                         else:
234                             if (plotting): plt.plot(np.array((discListX [point],
                        rayMirrorPointX2)), np.array((discListY [point],
                        rayMirrorPointY2)), color="lightgreen")
235                             if (checkIfLineEntersClearZone(np.array((discListX [
                        point], rayMirrorPointX2)), np.array((discListY [point
                        ], rayMirrorPointY2)), clearZoneX, clearZoneY,
                        showBadRays)):
236                                 badRays += 1
237
238         return badRays
239
240 #Optimizes an edge to ensure the minimum amount of rays enter the quite zone
241 def optimizeEdge(startPointX, startPointY, maxLength, orientation, height, srcX,
242                 srcY, clearZoneX, clearZoneY, screenX, screenY, discreteSize):
243     bestBadRays = -1
244     bestEdgeX = -1
245     bestEdgeY = -1
246
247     for i in np.linspace(0, maxLength-0.0001, 10):
248         #Define edge
249         if (orientation == 0):
250             edgeX = np.array((startPointX, startPointX+maxLength-i))
251             edgeY = np.array((startPointY, startPointY+height))
252         else:
253             edgeX = np.array((startPointX, startPointX+height))
254             edgeY = np.array((startPointY, startPointY-maxLength+i))
255
256         #Calculate the amount of bad rays with the current edge
257         badRays = findBadRays(srcX, srcY, edgeX, edgeY, clearZoneX, clearZoneY,
                screenX, screenY, discreteSize, False)
258         badRays += findBadRays(srcX, -srcY, edgeX, edgeY, clearZoneX, clearZoneY,
                screenX, screenY, discreteSize, False)
259
260         if (badRays == 0):
261             return edgeX, edgeY
262         elif (badRays < bestBadRays or bestBadRays == -1):
263             bestBadRays = badRays
264             bestEdgeX = edgeX
265             bestEdgeY = edgeY
266
267     return bestEdgeX, bestEdgeY
268
269 def optimizeSerratedPlate():
270     c0 = 3e8

```

```

270     fLow = 2e9
271     lamb = c0/fLow
272
273     #Define the size of the plate, clear zones and transmitter location
274     srcX = 0
275     srcY = 1.375/2 - 0.1
276     discreteSize = 0.05
277     screenX = 4
278     screenY = 4
279     clearZoneX = np.array((-0.55/2,0.55/2,0.55/2,-0.55/2,-0.55/2))
280     clearZoneY = np.array((1.325/2,1.325/2,-1.325/2,-1.325/2,1.325/2))
281     edgeLength = lamb*5
282
283     #Optimize each edge, and pass the starting point to the next optimization
284     leng = 1.375
285     width = 0.6
286     adjustedLength = 0
287     structureX = []
288     structureY = []
289     maxLengthX = lamb*1.5
290     maxLengthY = lamb*1.5
291     structureX.append(0)
292     structureY.append(leng/2)
293
294     running = True
295     placingTop = True
296     while (running):
297         if(placingTop):
298             if(len(structureX)%2 == 1):
299                 height = edgeLength
300             else:
301                 height = -edgeLength
302                 adjustedLength = 0
303                 length = width - np.abs(structureX[-1]) - adjustedLength
304                 if(length > maxLengthX):
305                     length = maxLengthX
306                 edgeX,edgeY = optimizeEdge(structureX[-1],structureY[-1],length,0,
307                     height,srcX,srcY,clearZoneX,clearZoneY,screenX,screenY,
308                     discreteSize)
309
310                 structureX.append(edgeX[1])
311                 structureY.append(edgeY[1])
312                 print("Points_placed: "+str(len(structureX))+ " X: "+str(structureX
313                     [-1])+ " Y: "+str(structureY[-1]))
314
315                 if structureX[-1] >= width/2 and height <= 0:
316                     placingTop = False
317                 elif structureX[-1] >= width/2 and height > 0:
318                     structureX.pop()
319                     structureY.pop()
320                     adjustedLength += (width/2 - np.abs(structureX[-1]))/100
321                     print("Point_Removed")
322             else:
323                 if(len(structureX)%2 == 1):
324                     height = edgeLength
325                 else:
326                     height = -edgeLength
327                     adjustedLength = 0
328                     length = np.abs(structureY[-1]) - adjustedLength
329                     if (length > maxLengthY):
330                         length = maxLengthY
331                     edgeX,edgeY = optimizeEdge(structureX[-1],structureY[-1],length,1,
332                         height,srcX,srcY,clearZoneX,clearZoneY,screenX,screenY,
333                         discreteSize)

```



```

331         structureX.append(edgeX[1])
332         structureY.append(edgeY[1])
333         print("Points_placed: "+str(len(structureX))+" X: "+str(structureX
334             [-1])+" Y: "+str(structureY[-1]))
335
336         if structureY[-1] <= 0 and height <= 0:
337             running = False
338         elif structureY[-1] <= 0 and height > 0:
339             structureX.pop()
340             structureY.pop()
341             adjustedLength += (leng - np.abs(structureY[-1]))/100
342             print("Point_Removed")
343
344         #Mirror this section to the other 3 quadrants
345         noOfWedges = (len(structureX)-1)/2
346         for i in np.arange(0, len(structureX)-1):
347             structureX.append(structureX[-2*i-2])
348             structureY.append(-1*structureY[-2*i-2])
349         for i in np.arange(0, len(structureX)-1):
350             structureX.append(-structureX[-2*i-2])
351             structureY.append(structureY[-2*i-2])
352
353         findBadRays(srcX, srcY, structureX, structureY, clearZoneX, clearZoneY, screenX,
                    screenY, discreteSize, True)

```

Appendix E

Multi-copter Software, Antennas and Specification Sheets

E.1 RFM22B daughter board

The schematic for the PCB, designed to attach the RFM22B module to the Raspberry-Pi, can be seen in Fig. E.1. The PCB itself was designed as a four-layer board. The middle two layers were used as the supply and ground power planes. Additionally, the top and bottom signal path layers were also filled with a stitched ground-plane to minimise RFI coupling. The four layers of the PCB can be seen in Fig. E.2. The trace between the SMA connector and the input pin of the transceiver was specified as a $50\ \Omega$ transmission line. This meant that the trace needed to have a thickness of 0.3 mm given the PCB fabrication specifications in Table E.1.

Top-Layer Copper Height	1.4 mil
Top-Layer Substrate Height	6.7 mil
2nd Layer Copper Height	0.7 mil
Core Substrate Height	47 mil
3rd Layer Copper Height	0.7 mil
3rd Layer Substrate Height	6.7 mil
Bottom-Layer Copper Height	1.4 mil
Substrate Dielectric	3.66 (1 GHz)

Table E.1: Fabrication specifications for a 4-layer PCB from OSH Park [88].

APPENDIX E. MULTI-COPTER SOFTWARE, ANTENNAS AND SPECIFICATION SHEETS

152

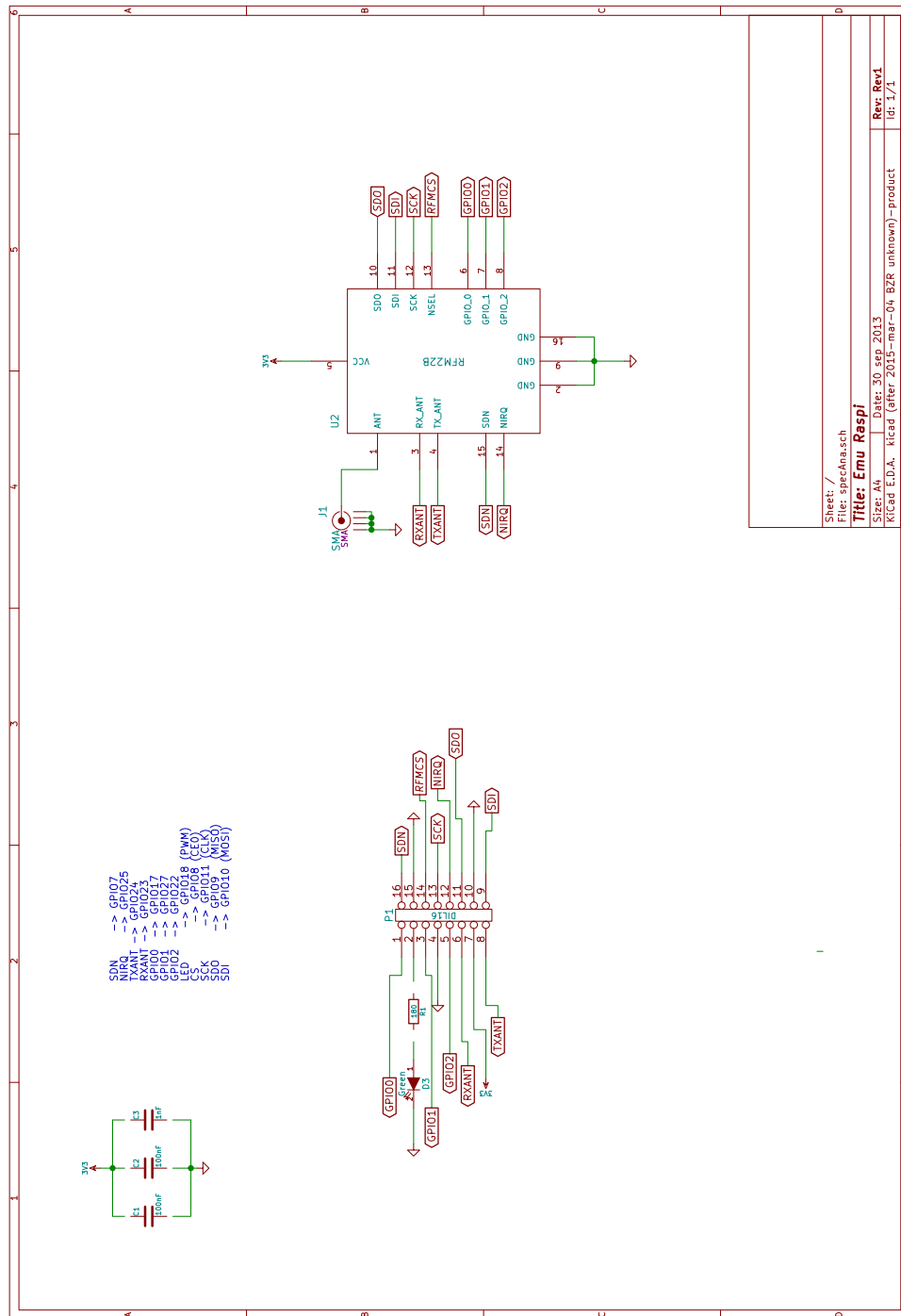


Figure E.1: Daughter board schematic, drawn using KiCad [87].

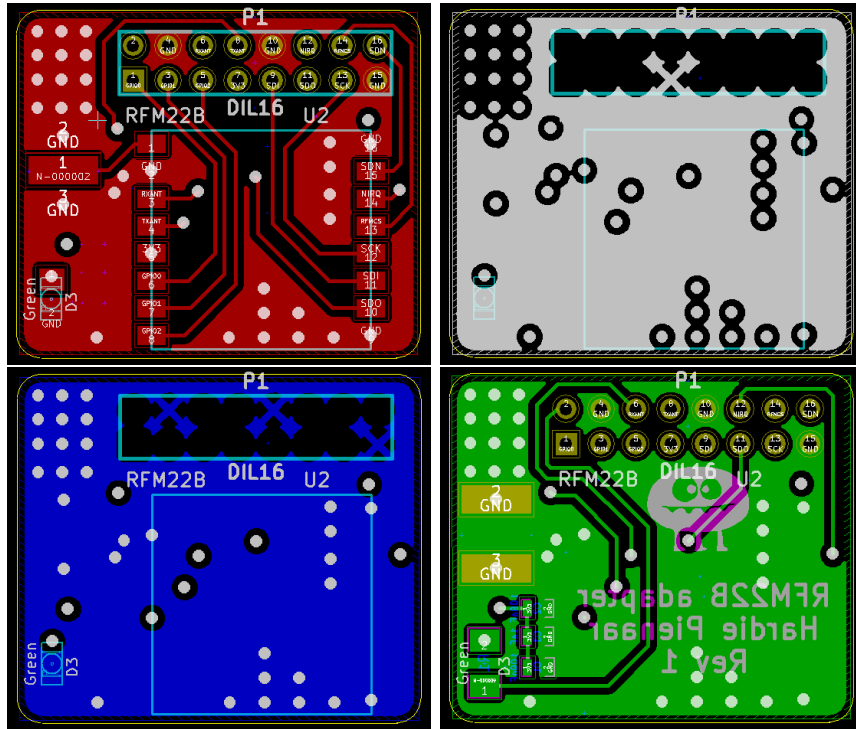
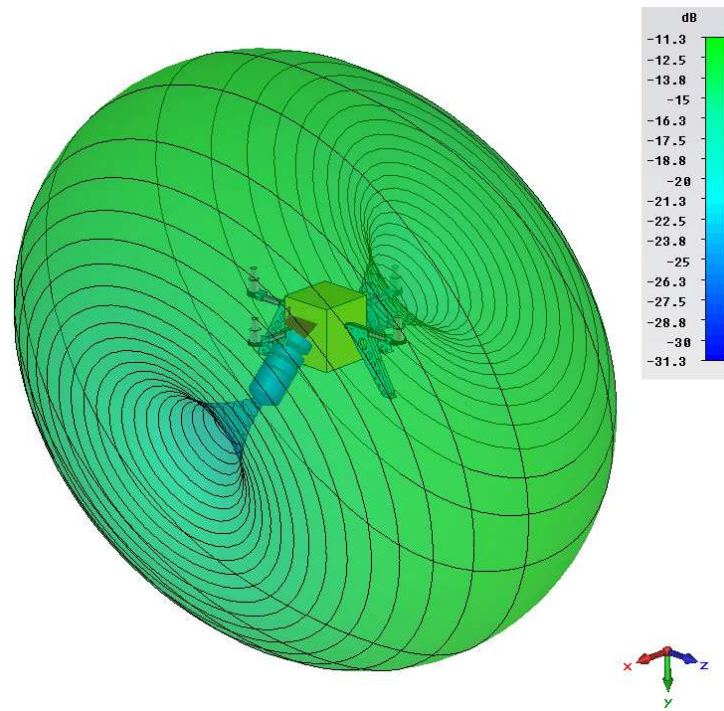


Figure E.2: PCB design for the RFM22B daughter board. The PCB is responsible for connecting the transceiver module to the Raspberry-Pi. Additionally, an SMA connector for the antenna is supplied.

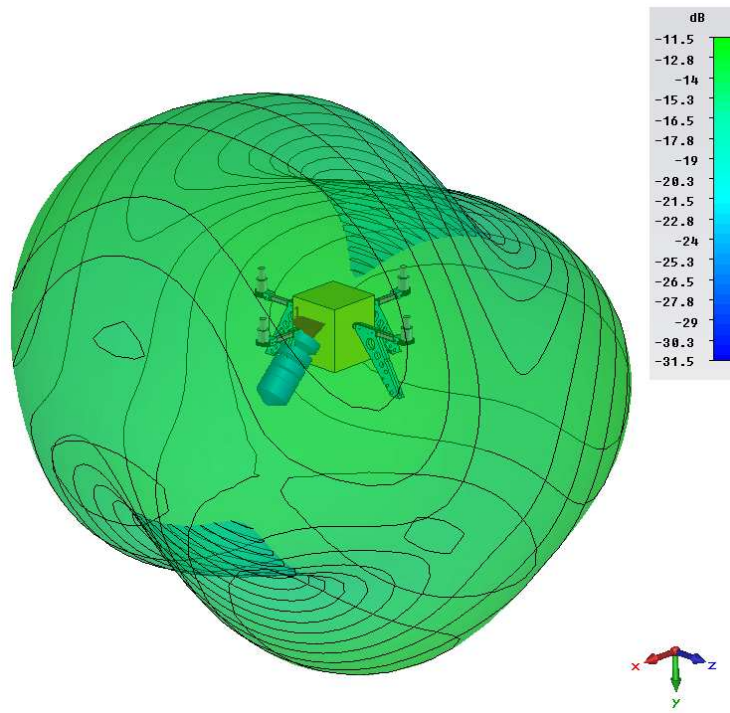
E.2 3D-Antenna Patterns

Fig E.3 shows the Multi-copter antenna patterns simulated in CST[®]. These simulations take into account most of the metallic and dielectric structures. Each antenna pattern represents one leg-antenna only. The full-isotropic antenna described in the dissertation is a combination of two 90 degree offset elements. The three discrete-simulated frequencies represent the bandwidth span for the loaded elements.

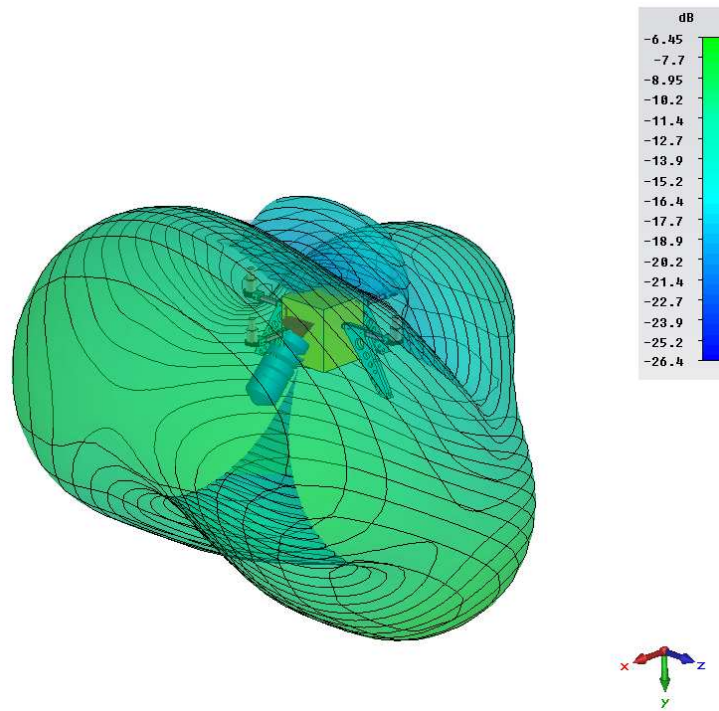


(a) 260 MHz

Figure E.3: Multi-copter far-field antenna patterns. These realised-gain patterns give an indication of the field at three discrete frequency points. Continued in E.4.



(b) 550 MHz



(c) 900 MHz

Figure E.4: Multi-copter far-field antenna patterns. These realised-gain patterns give an indication of the field at three discrete frequency points.

E.3 Specifications sheets

DC-Brushless motor specifications from the manufacturer website can be seen in Fig. E.5. These specifications give performance details for different supply voltages and propeller-pitch angles. In this project, an 11.1 V battery with a 12x4.5 pitch propeller was used. The electrical specifications and performance graphs for the ZX60-P105LN+ can be seen in Fig. E.6 and Fig. E.7 respectfully.

Item No.	Volts (V)	Prop	Throttle	Amps (A)	Watts (W)	Thrust (G)	RPM	Efficiency (G/W)	Operating temperature(°C)
MT2216 KV800	11.1	T-MOTOR 10*3.3CF	50%	2.3	27	290	4800	10.74	37
			65%	3.1	36	320	5500	8.89	
			75%	4	45	380	6000	8.44	
			85%	5.1	58	470	6500	8.10	
			100%	6.2	69	550	7000	7.97	
		T-MOTOR 11*3.7CF	50%	2.6	30	330	4500	11.00	39
			65%	3.7	42	350	5200	8.33	
			75%	4.7	53	420	5700	7.92	
			85%	6.3	70	540	6200	7.71	
			100%	7.4	83	620	6650	7.47	
		T-MOTOR 12*4CF	50%	2.9	34	380	3900	11.18	42
			65%	4.9	55	480	4800	8.73	
			75%	6.6	73	590	5200	8.08	
			85%	8.4	91	690	5600	7.58	
			100%	9.9	107	800	6000	7.48	
	14.8	T-MOTOR 9*3CF	50%	2.9	43	330	6800	7.67	42
			65%	3.6	55	360	7400	6.55	
			75%	4.5	67	420	8000	6.27	
			85%	6.1	90	540	8700	6.00	
			100%	7.1	105	620	9300	5.90	
		T-MOTOR 10*3.3CF	50%	3.4	52	430	6200	8.27	45
			65%	4.7	71	520	6800	7.32	
			75%	6	90	610	7500	6.78	
			85%	7.9	117	750	8300	6.41	
			100%	9.5	140	850	8800	6.07	
		T-MOTOR 11*3.7CF	50%	3.7	55	450	5700	8.18	48
			65%	5.6	82	580	6600	7.07	
			75%	7.2	106	700	7200	6.60	
			85%	9.3	139	900	7800	6.47	
			100%	11.1	164	1020	8300	6.22	

Figure E.5: Specification sheet for the MT2216 (800 KV) dc-brushless motor [89].

APPENDIX E. MULTI-COPTER SOFTWARE, ANTENNAS AND SPECIFICATION SHEETS

157

Parameter	Condition (MHz)	Min.	Typ.	Max.	Units
Frequency Range		40		2600	MHz
Noise Figure	40 500 900 2000 2600		2.3 2.0 1.9 1.9 2.0	2.7	dB
Gain	40 500 900 2000 2600	13.8	14.4 14.5 14.4 15.5 15.1	16.8	dB
Gain Flatness	1000 - 2000		±0.25		dB
Output Power @ 1 dB compression	40 500 900 2000 2600		19.5 21.0 21.0 18.9 19.4		dBm
Output IP3	40 500 900 2000 2600		34.6 38.7 37.4 33.6 33.2		dBm
Input VSWR	40 500 900 2000 2600		2.2 1.2 1.2 1.3 1.8		dB
Output VSWR	40 500 900 2000 2600		1.1 1.2 1.1 2.4 2.2		dB
Active Directivity (Isolation-Gain)	40 500 900 2000 2600		6.3 4.5 5.1 8.1 13.5		dB
DC Supply Voltage		4.8	5.0	5.2	V
Supply Current		—	63	77	mA

Figure E.6: Electrical Specifications of the ZX60-P105LN+ from Mini-Circuits.

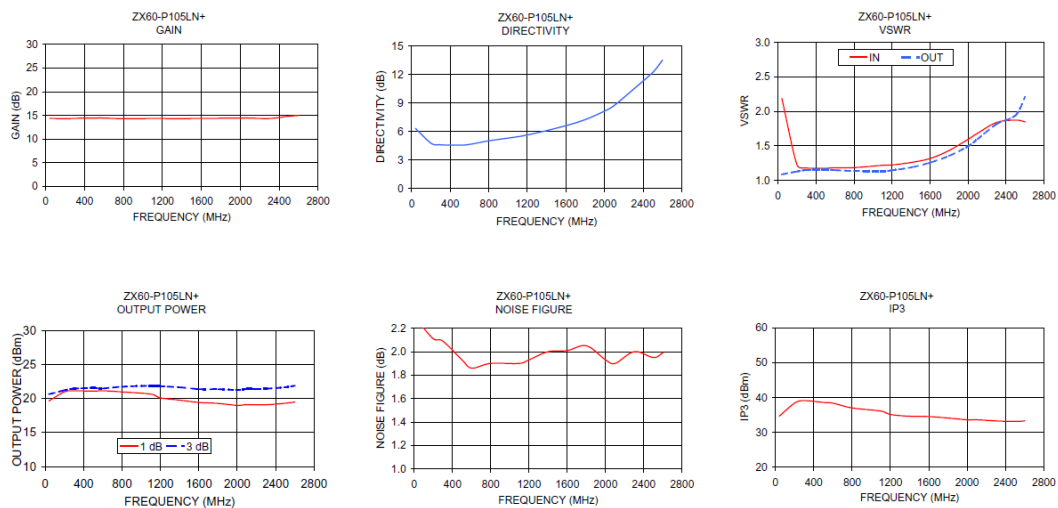


Figure E.7: Performance Graphs of the ZX60-P105LN+ from Mini-Circuits.

E.4 Raspberry-Pi logging software

During flight, a python script was used to log the measurements continuously. This script was executed on the Raspberry-Pi itself and can be seen in Listing E.1. Additionally, the driver code can be seen in Listing E.2. The logging script starts by importing all of its necessary libraries followed by the initialisation of its global variables. The main part of the program then initiates the sensor and telemetry threads. These threads are responsible for reading and making its corresponding data available. After starting these threads, a file handle is opened, and the main loop is started. The main loop continuously checks if new information is available. As soon as data is ready, it is stored in the opened file.

Listing E.1: Raspberry-Pi Measurement Logging Script

```

1 import time
2 import os,struct,sys
3 import RFM22BRaspi as rfm
4
5
6 #Threading switches, used to turn threads off at the end
7 #of their respective loops
8 flightLoggerRunning = True
9 sensorLoggerRunning = True
10 guiRunning = True
11 mainThreadRunning = True
12
13 #Mutexes
14 flightDataMutex = threading.Lock()
15 sensorDataMutex = threading.Lock()
16
17 #Thread sleep times in seconds
18 flightLoggerSleep = 0.001
19 sensorLoggerSleep = 0.001
20 guiSleep = 3
21 mainLoopSleep = 0.001
22
23 #Flight Data Variables
24 missionSeq = -999
25 batVoltage = -999
26 batCurrent = -999
27 batRemaining = -999
28
29 roll = -999
30 pitch = -999
31 yaw = -999
32
33 heading = -999
34 barAlt = -999
35 pressure = -999
36 temp = -999
37
38 noOfSats = -999
39 gpsFix = -999
40 gpsAlt = -999
41 lat = -999
42 lon = -999
43 timeStamp = -999
44
45 #Sensor Data Variables
46 level0Max = -999

```

APPENDIX E. MULTI-COPTER SOFTWARE, ANTENNAS AND SPECIFICATION SHEETS

159

```

47 level0Min = -999
48 level0Ave = -999
49 level1Max = -999
50 level1Min = -999
51 level1Ave = -999
52
53 #Control switches
54 verboseThreading = False
55 verboseAPM = False
56 showGUI = True
57 newDataAvailable = False
58
59 #Sensor settings
60 frequency = 430.0
61 rbw = 21000
62 preAmp = True
63 dwell = 100
64
65 #File settings
66 filename = "KAPBRoom1430.csv"
67
68 ##### Flight Logger Thread
69
70 def FlightLogger():
71     global flightLoggerRunning
72
73     #Link global variables
74     global missionSeq
75     global batVoltage
76     global batCurrent
77     global batRemaining
78
79     global roll
80     global pitch
81     global yaw
82
83     global heading
84     global barAlt
85     global pressure
86     global temp
87
88     global noOfSats
89     global gpsFix
90     global gpsAlt
91     global lat
92     global lon
93     global timeStamp
94
95     #Setup Connection with APM
96     '''
97     Backend of UAV data collection.
98     This class will interface via a serial link
99     on an MAVLink protocol to a UAV drone to extract
100     the necessary data and make it available for
101     external use. This class will make extensive use
102     of the already existing pymavlink software from Lorenz Meier.
103     github.com/mavlink/pymavlink
104     '''
105
106
107     setStreamRate on an APM
108
109     '''
110     os.path.join(os.path.dirname(os.path.realpath(__file__)), '..')
111
112     from optparse import OptionParser

```

APPENDIX E. MULTI-COPTER SOFTWARE, ANTENNAS AND SPECIFICATION SHEETS

160

```

113     parser = OptionParser("apmsetrate.py_[options]")
114
115     parser.add_option("--baudrate", dest="baudrate", type='int',
116                      help="master_port_baud_rate", default=115200)
117     parser.add_option("--device", dest="device", default=None, help="serial_
118                        device")
119     parser.add_option("--rate", dest="rate", default=4, type='int', help="
120                        requested_stream_rate")
121     parser.add_option("--source-system", dest='SOURCE_SYSTEM', type='int',
122                      default=255, help='MAVLink_source_system_for_this_GCS')
123     parser.add_option("--showmessages", dest="showmessages", action='
124                        store_true',
125                      help="show_incoming_messages", default=False)
126     (opts, args) = parser.parse_args()
127
128     opts.device = "/dev/ttyACM0"
129     #opts.device = "/dev/ttyUSB0"
130
131     import mavutil
132
133     if opts.device is None:
134         if(verboseAPM): print("You_must_specify_a_serial_device")
135         sys.exit(1)
136
137     def wait_heartbeat(m):
138         '''wait_for_a_heartbeat_so_we_know_the_target_system_IDS'''
139         m.wait_heartbeat()
140
141     def get_messages(m):
142         '''show_incoming_mavlink_messages'''
143         while True:
144             msg = m.recv_match(blocking=True)
145             if not msg:
146                 return 'NULL'
147             if msg.get_type() == "BADDATA":
148                 if mavutil.all_printable(msg.data):
149                     sys.stdout.write(msg.data)
150                     sys.stdout.flush()
151             else:
152                 return msg
153
154     # create a mavlink serial instance
155     master = mavutil.mavlink_connection(opts.device, baud=opts.baudrate)
156
157     # wait for the heartbeat msg to find the system ID
158     wait_heartbeat(master)
159
160     while(flightLoggerRunning):
161         flightDataMutex.acquire()
162
163         message = str(get_messages(master))
164         #Check type of message
165         if('SCALED_PRESSURE' in message):
166             data = message.split(':')
167             pressure = float(data[2].split(',')[0])
168             temp = int(data[4].split('}')[0])
169         elif('VFR_HUD' in message):
170             data = message.split(':')
171             barAlt = float(data[5].split(',')[0])
172             heading = int(data[3].split(',')[0])
173         elif('ATTITUDE' in message):
174             data = message.split(':')
175             roll = float(data[2].split(',')[0])
176             pitch = float(data[3].split(',')[0])

```

APPENDIX E. MULTI-COPTER SOFTWARE, ANTENNAS AND SPECIFICATION SHEETS

161

```

176         yaw = float(data[4].split(',')[0])
177     elif('GLOBALPOSITION.INT' in message):
178         data = message.split(':')
179         timeStamp = int(data[1].split(',')[0])
180         lat = float(data[2].split(',')[0])
181         lon = float(data[3].split(',')[0])
182         gpsAlt = float(data[4].split(',')[0])
183     elif('GPS.RAW.INT.' in message):
184         data = message.split(':')
185         noOfSats = int(data[10].split(',')[0])
186         gpsFix = int(data[2].split(',')[0])
187     elif('MISSION.CURRENT' in message):
188         data = message.split(':')
189         missionSeq = int(data[1].split(',')[0])
190     elif('SYS.STATUS' in message):
191         data = message.split(':')
192         batVoltage = int(data[5].split(',')[0])
193         batCurrent = int(data[6].split(',')[0])
194         batRemaining = int(data[7].split(',')[0])
195     else:
196         flightDataMutex.release()
197
198     #Rest thread
199     time.sleep(flightLoggerSleep)
200
201
202     #Close down serial link when thread exits
203     master.close()
204
205
206     ##### End of Flight Logger Thread
207
208     ##### Sensor Logger Thread
209
210 def SensorLogger():
211     #Initialize sensor logger global variables
212     global level0Max
213     global level0Min
214     global level0Ave
215     global level1Max
216     global level1Min
217     global level1Ave
218     global sensorLoggerRunning
219     global frequency
220     global newDataAvailable
221
222     rfm.initializeGPIO()
223     spi0 = rfm.initializeRFM22B(False,0)
224     spi1 = rfm.initializeRFM22B(False,1)
225
226     #Check and handle a non-connection
227     if(not rfm.checkRFM22BStatus(spi0) or not rfm.checkRFM22BStatus(spi1)):
228         print("Connection to sensor failed")
229         killProgram()
230     rfm.setFrequency(spi0,frequency)
231     rfm.setFrequency(spi1,frequency)
232     rfm.setRBW(spi0,rbw)
233     rfm.setRBW(spi1,rbw)
234     rfm.setGain(spi0,preAmp)
235     rfm.setGain(spi1,preAmp)
236
237     while(sensorLoggerRunning):
238         rfm.setLED(True)
239         sensorDataMutex.acquire()
240         level0Min,level0Ave,level0Max,noOfMeas = rfm.getRSSI(spi0,dwell)
241         level1Min,level1Ave,level1Max,noOfMeas = rfm.getRSSI(spi1,dwell)

```

APPENDIX E. MULTI-COPTER SOFTWARE, ANTENNAS AND SPECIFICATION SHEETS

162

```

242         newDataAvailable = True
243         sensorDataMutex.release()
244         rfm.setLED(False)
245         #Rest thread
246         time.sleep(sensorLoggerSleep)
247
248 ##### End of Sensor Logger Thread
249
250
251 ##### General functions
252
253 def killProgram():
254     global flightLoggerRunning
255     global sensorLoggerRunning
256     global guiRunning
257     global verboseThreading
258     global sLThread
259     global fLThread
260     global guiThread
261     global file
262
263     #Send stop signal to all threads
264     flightLoggerRunning = False
265     sensorLoggerRunning = False
266     guiRunning = False
267
268     #Wait for threads to stop before doing final operations
269     while(sLThread.isAlive() or fLThread.isAlive() or guiThread.isAlive()):
270         time.sleep(0.5)
271
272     #Close file
273     file.close()
274
275     #Shut down program
276     sys.exit()
277
278
279 ##### End of General function
280
281
282 #Create main threads
283 fLThread = threading.Thread(target = FlightLogger)
284 sLThread = threading.Thread(target = SensorLogger)
285 guiThread = threading.Thread(target = GUI)
286
287 #Start main threads
288 fLThread.start()
289 sLThread.start()
290 guiThread.start()
291
292 #Run main program
293 file = open(filename, 'w')
294
295 #Write file header
296 file.write("fEMu_2.0_flight_signal_file\n")
297 file.write("Frequency_[MHz]: , "+str(frequency)+"\n")
298 file.write("RBW_[Hz]: , "+str(rbw)+"\n")
299 file.write("Dwell_[ms]: , "+str(dwell)+"\n")
300 file.write("time,No_of_sats,GPS_fix,Lat,Lon,GPS_Alt,Barometer_Alt,Heading,Roll
    ,Pitch,Yaw,Abs_Pressure,Temperature,Mission_Seq,Battery_Voltage,Battery_
    Remaining,Battery_Current,RSSI_0_Max,RSSI_0_Ave,RSSI_0_Min,RSSI_1_Max,RSSI
    _1_Ave,_RSSI_1_Min\n")
301
302 try:
303     while(mainThreadRunning):
304         #Check if new data is available

```

APPENDIX E. MULTI-COPTER SOFTWARE, ANTENNAS AND SPECIFICATION SHEETS

163

```

305         if(newDataAvailable):
306             flightDataMutex.acquire()
307             sensorDataMutex.acquire()
308             file.write(str(timestamp)+","+str(noOfSats)+","+str(gpsFix)+","+
                        str(lat)+","+str(lon)+","+str(gpsAlt)+","+str(barAlt)+","+str(
                        heading)+","+str(roll)+","+str(pitch)+","+str(yaw)+","+str(
                        pressure)+","+str(temp)+","+str(missionSeq)+","+str(batVoltage
                        )+","+str(batRemaining)+","+str(batCurrent)+","+str(level0Max)
                        +","+str(level0Ave)+","+str(level0Min)+","+str(level1Max)+","+
                        str(level1Ave)+","+str(level1Min)+"\n")
309             newDataAvailable = False
310             flightDataMutex.release()
311             sensorDataMutex.release()
312
313
314             #Give an error if a system could not connect
315             #Attempt a reconnect after a disconnection
316             #Only start logging if all systems are on and connected
317             #Check if any new data is available, if so, store in log
318             #Implement sensor logger
319             time.sleep(mainLoopSleep)
320 except (KeyboardInterrupt, SystemExit):
321     print("Keyboard_Interrupt")
322     killProgram()

```

Listing E.2: Driver code for RFM22B

```

324 import numpy as np          #For all the number crunching
325 import matplotlib.pyplot as plt #For the nice plots
326 import time as t           #For keeping time between events
327 import csv as csv          #For writing to files
328 import RPi.GPIO as GPIO    #For working with GPIO header
329 import spidev as spidev    #For working with spi channels
330
331
332 #Global variables
333 #Port allocations
334 LEDPort = 18
335 NIRQPort = 25
336 TXAntPort = 24
337 RXAntPort = 23
338 GPIO0Port = 17
339 GPIO1Port = 27
340 GPIO2Port = 22
341
342 #IF Filter lookup table
343 IFFilter = [2600,2800,3100,3200,3700,4200,4500,4900,5400,
344 5900,6100,7200,8200,8800,9500,10600,11500,12100,14200,16200,
345 17500,18900,21000,22700,24000,28200,32200,34700,37700,41700,
346 45200,47900,56200,64100,69200,75200,83200,90000,95300,112100,
347 127900,137900,142800,167800,181100,191500,225100,248800,269300,
348 284900,335500,361800,420200,469400,518800,577000,620700]
349
350 IFFilterNDecExp = [5,5,5,5,5,5,5,4,4,4,4,4,4,3,3,3,3,3,3,
351 2,2,2,2,2,2,2,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,
352 0,0,0,0,0,0]
353
354 IFFilterDwn3Bypass = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
355 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,
356 1,1,1,1,1,1]
357
358 IFFilterFilset = [1,2,3,4,5,6,7,1,2,3,4,5,6,7,1,2,3,4,5,6,7,1,2,
359 3,4,5,6,7,1,2,3,4,5,6,7,1,2,3,4,5,6,7,4,5,9,15,1,2,3,4,8,9,10,11,
360 12,13,14]
361
362 #Initializes GPIO header pins

```

APPENDIX E. MULTI-COPTER SOFTWARE, ANTENNAS AND SPECIFICATION SHEETS

164

```

363 def initializeGPIO():
364     #Turn off GPIO warnings
365     GPIO.setwarnings(False)
366
367     #Set GPIO mode
368     GPIO.setmode(GPIO.BCM)
369
370     #RFM22B GPIO Ports needs to be allocated below
371     #Set output ports
372     GPIO.setup(LEDPort,GPIO.OUT)
373     GPIO.setup(TXAntPort,GPIO.OUT)
374     GPIO.setup(RXAntPort,GPIO.OUT)
375
376     #Set input ports
377     GPIO.setup(NIRQPort,GPIO.IN)
378
379     #Turns LED on or off according to given state
380     #state = True: LED will turn on
381     #state = False: LED will turn off
382     def setLED(state):
383         if state:
384             GPIO.output(LEDPort,True)
385         else:
386             GPIO.output(LEDPort,False)
387
388     #Sets up the RFM22B to a initialized state
389     #Returns the spi communication handle
390     def initializeRFM22B(verbose=True,channel=0):
391         #Put the RFM22B in a RX State
392         GPIO.output(TXAntPort,False)
393         GPIO.output(RXAntPort,True)
394         t.sleep(0.1) #Wait for device to assume state
395
396         #Initialize SPI channel
397         spi = spidev.SpiDev() #Creates spi object
398         spi.open(0,channel) #Opens channel with CE0 as enable port
399
400         #Turn on PLL to enter tune mode
401         writeRFM22BRegister(spi,0x07,0b00000110)
402
403         #Turn off agc and set LNA gain to 5dB minimum and pga gain to 0
404         writeRFM22BRegister(spi,0x69,0b00000000)
405
406         return spi #Return a handle to the spi communication handle
407
408     #Returns the value at a specific address on the RFM22B
409     #address must be specified in hex eg. 0x0A
410     #spi is the spi communication channel which must be configured beforehand
411     def readRFM22BRegister(spi,address):
412         resp = spi.xfer2([address,0x00])
413         return resp[1]
414
415     #Writes the given data to the given address on the RFM22B
416     #address must be specified in hex eg. 0x0A
417     #spi is the spi communication channel which must be configured beforehand
418     #data is the 8 bit data in hex format that needs to be stored at the address
419     def writeRFM22BRegister(spi,address,data):
420         #Add write bit to address
421         address += 128
422         #Send address and data to be written
423         resp = spi.xfer2([address,data])
424
425     #Prints out the current status of the RFM22B module
426     #If verbose is true the output will be written to the console
427     #Returns True if module is working
428     #Returns False if module is not working

```

APPENDIX E. MULTI-COPTER SOFTWARE, ANTENNAS AND SPECIFICATION SHEETS

165

```

429 def checkRFM22BStatus(spi, verbose=True):
430     #Read the device type, version and status
431     deviceType = readRFM22BRegister(spi, 0x00)
432     deviceVersion = readRFM22BRegister(spi, 0x01)
433     deviceStatus = readRFM22BRegister(spi, 0x02)
434
435     #Check if the device type is read correctly, this will determine if
436     communication is and the device is working
437     if(deviceType != 8): #Check if device is not of tranceiver type or if
438         communication is broken
439         return False
440     return True
441
442 #Sets up the RFM22B registers to listen or transmit at the given frequency
443 #Returns True if the command was successful
444 #freq is specified in MHz
445 def setFrequency(spi, freq):
446     #Check if frequency is in range
447     if(freq > 960 or freq < 240):
448         return False
449
450     #Some variables that will be used
451     bandSelect = 0b01000000
452     fb = 0b00000000
453     fc = 0
454     xtalFreq = 30000
455
456     #Calculate register according to the formula found in excel sheet
457     if(freq >= 480):
458         bandSelect |= 0b00100000
459         temp = freq/(10*(xtalFreq/30000)*(1+1))
460     else:
461         temp = freq/(10*(xtalFreq/30000)*(1+0))
462     fb = int(np.floor(temp)) - 24
463     fc = int(np.floor((temp - np.floor(temp))*64000+0.49999))
464
465     bandSelect |= fb
466     fc1 = fc>>8
467     fc2 = fc&0b0000000011111111
468
469     #Write the calculated registers to the device
470     writeRFM22BRegister(spi, 0x75, bandSelect)
471     writeRFM22BRegister(spi, 0x76, fc1)
472     writeRFM22BRegister(spi, 0x77, fc2)
473
474     return True
475
476 #Sets the IF filter register for the specified bandwidth by using a lookup
477 table
478 #rbw is the bandwidth in Hz
479 #returns True if filter was found in table and set
480 def setRBW(spi, rbw):
481     #Run through list to find IF chosen index
482     index = -1
483     for i in np.arange(0, len(IFFilter)):
484         if(int(rbw) == int(IFFilter[i])):
485             index = i
486
487     #Return False if rbw was not found in list
488     if index < 0:
489         return False
490
491     #Construct registers out of lookup table

```


APPENDIX E. MULTI-COPTER SOFTWARE, ANTENNAS AND SPECIFICATION SHEETS

166

```

492     ifReg = 0b00000000
493     ifReg |= (IFFilterDwn3Bypass[index]<<7)
494     ifReg |= (IFFilterNDecExp[index]<<4)
495     ifReg |= (IFFilterFilset[index])
496
497     #Write registers to RFM22B
498     writeRFM22BRegister(spi, 0x1C, ifReg)
499
500     return True
501
502 #Sets the gains in the RFM22B by using a lookup table for the pga gains and a
boolean for the LNA gain
503 #Returns True if the lookup table entry was found
504 def setGain(spi, lna):
505     lnaGain = -1
506     pgaGain = 0    #PGA Gain is not yet implemented
507
508     #set LNA to 25dB:
509     if lna:
510         lnaGain = 1
511     else: #Set LNA to 5dB
512         lnaGain = 0
513
514     #Create register value
515     gainReg = 0b00000000
516     gainReg |= lnaGain<<4
517     gainReg |= pgaGain
518
519     #Write register
520     writeRFM22BRegister(spi, 0x69, gainReg)
521
522     return True
523
524 #Reads, scales and calibrates the RSSI level from the RFM22B
525 #dwell specifies the amount of ms to measure
526 def getRSSI(spi, dwell):
527     maxValue = 0
528     minValue = 300
529     rawValue = 0
530     averageCount = 0
531
532     #Start averaging loop until dwell time is reached
533     startTime = t.time()
534     while t.time() - startTime <= dwell/1000.0 or averageCount == 0:
535         #Read 8 bit value from RFM22B
536         value = readRFM22BRegister(spi, 0x26)
537
538         #Store the minimum and maximum values
539         if (value > maxValue):
540             maxValue = value
541         if (value < minValue):
542             minValue = value
543
544         #build average
545         rawValue += value
546         averageCount += 1
547     rawValue = rawValue/averageCount
548
549     #Scale value to dBm
550     #This is only a raw scaling
551     RSSIAve = 0.5*rawValue - 120
552     RSSIMax = 0.5*maxValue - 120
553     RSSIMin = 0.5*minValue - 120
554
555     return RSSIMin, RSSIAve, RSSIMax, averageCount

```

Appendix F

Aerial Real-time Transient Analyser Development

F.1 In-lab power-line sparking measurement

In order to obtain a measure of the bandwidth and power associated with power-line sparking, in-lab measurements were done. A spark-gap was attached to the centre line of our rooftop high-voltage line. To simulate real-



Figure F.1: Photo showing the spark-gap on the 12 kV line. The LPDA receiving antenna can be seen as the orange structure directed at the spark-gap.

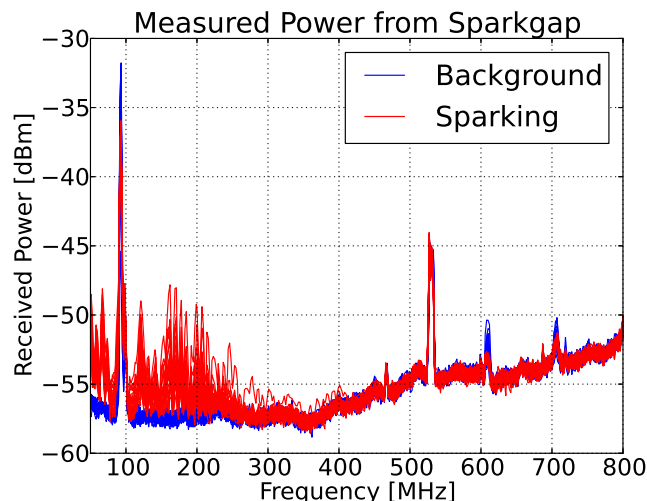


Figure F.2: Sparking measurement showing the broadband nature of the pulses extending to 400 MHz. A background measurement distinguishes the environment from sparking.

world conditions, a voltage of 12 kV was used. A wideband LPDA antenna connected to real-time transient analyser was used to measure the sparking emissions (see Fig. F.1). A total of 40 dB of attenuation was used to protect our front-end amplifiers. The measurement with attenuation de-embedded can be seen in Fig. F.2. A background characterisation was done before measurement to distinguish the sparking from the environmental noise.

F.2 Real-time transient analyser design

The design of the aerial real-time transient analyser involved a compact PCB layout along with a system level integration of the FPGA, ADC, clock, filterbank and single-board-computer components. This appendix includes the schematics, PCB layouts, high-level Python code as well as the low-level FPGA code that was used to realise the instrument.

The light-weight metrology device can instantaneously sample a bandwidth of 400 MHz up to the highest frequency of 700 MHz with a resolution of 8 bits. At a sample rate of 1 GSPS a total of 60 μ s can be captured at a resolution of 1 ns. This makes the instrument well-suited for a Multi-copter campaign where RFI in the form of transients is present. An example mentioned earlier in the appendix is sparking from HV power lines which was also the original intent of this device.

Schematics

Figure F.3: Root page

APPENDIX F. AERIAL REAL-TIME TRANSIENT ANALYSER
DEVELOPMENT

170

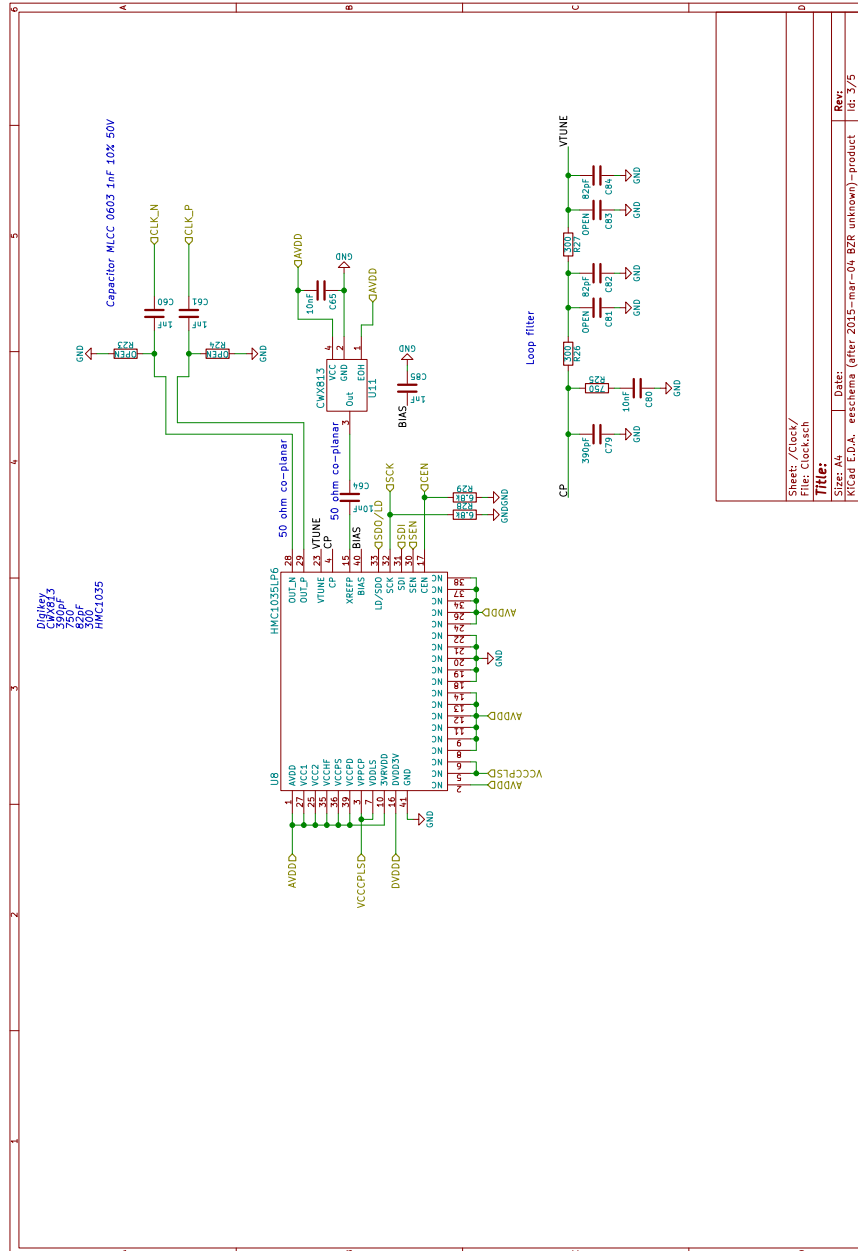


Figure F.4: Clock page

APPENDIX F. AERIAL REAL-TIME TRANSIENT ANALYSER DEVELOPMENT

171

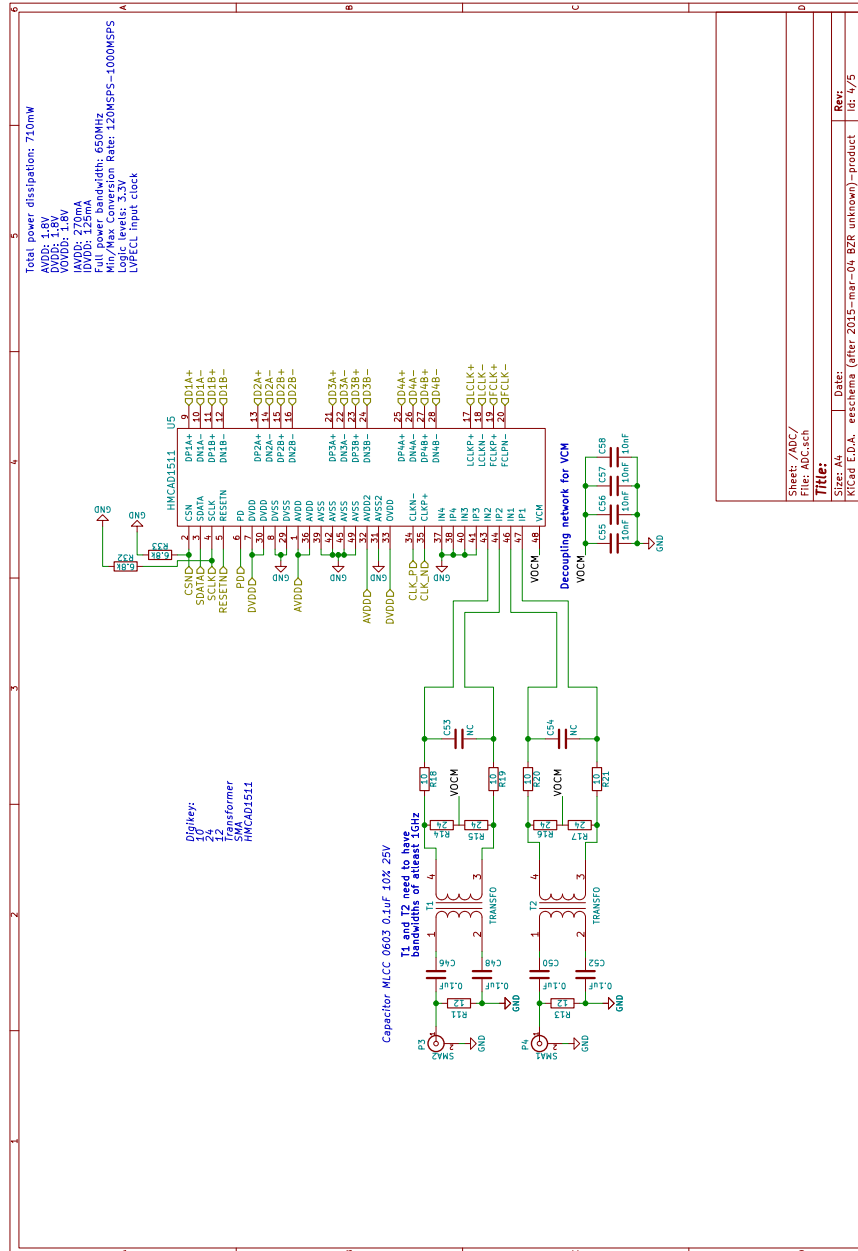


Figure F.5: ADC page

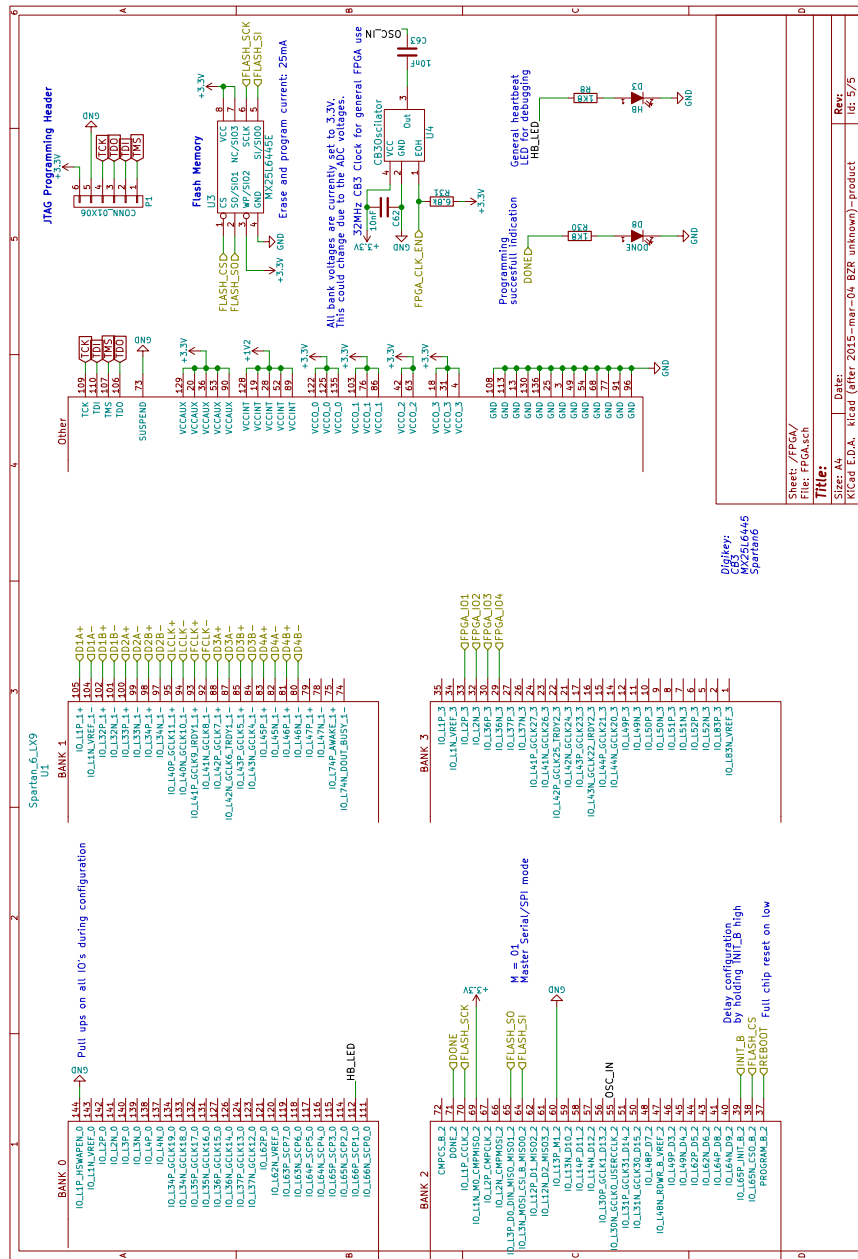


Figure F.6: FPGA page

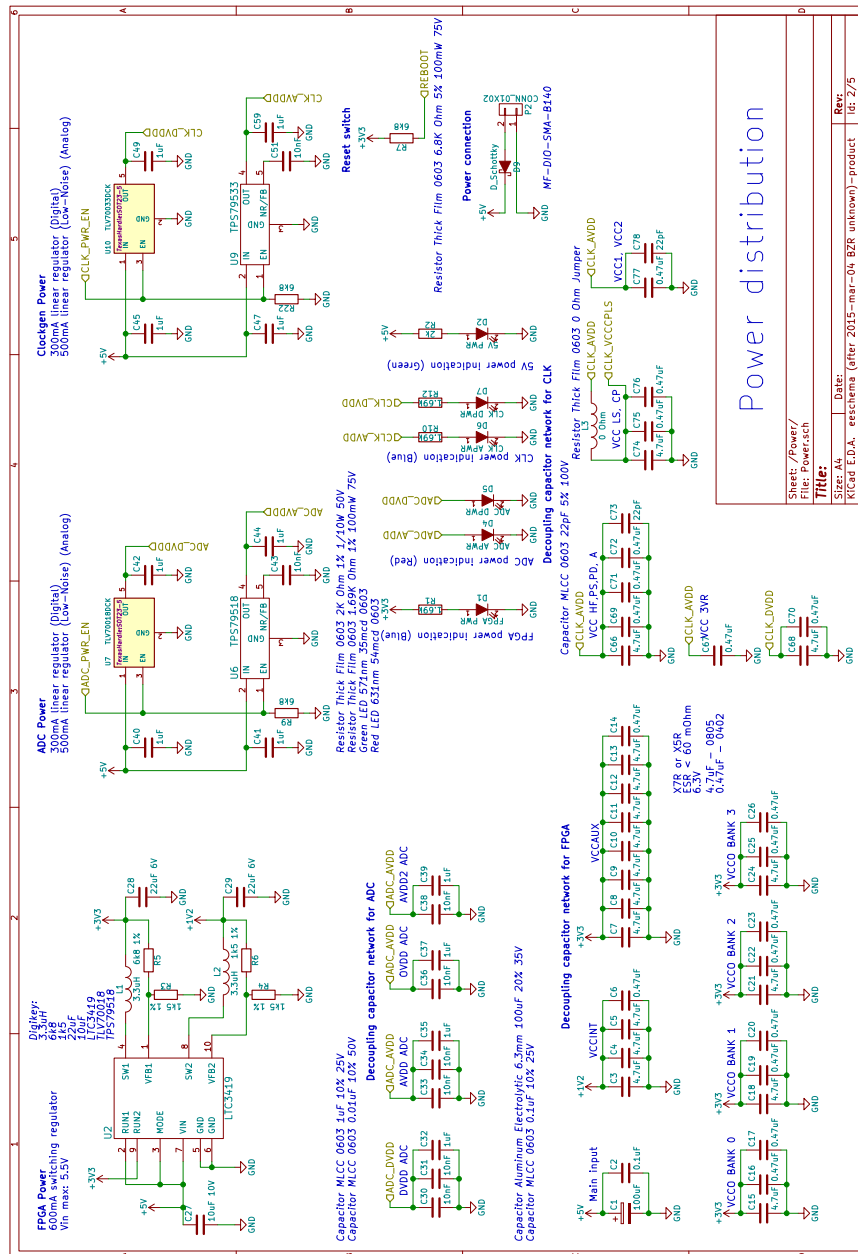


Figure F.7: Power page

PCB Layout

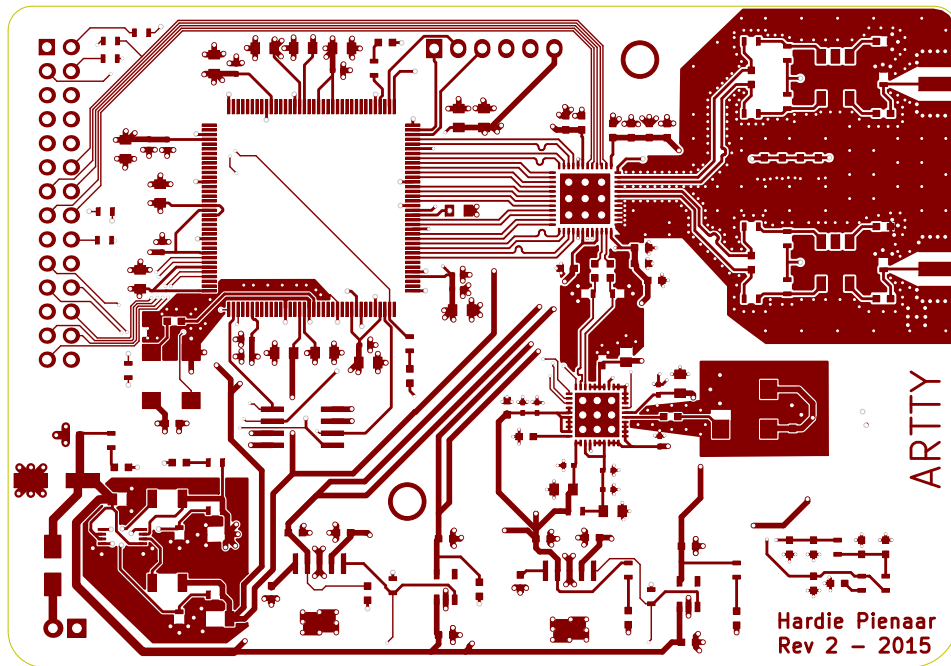


Figure F.8: Front Copper

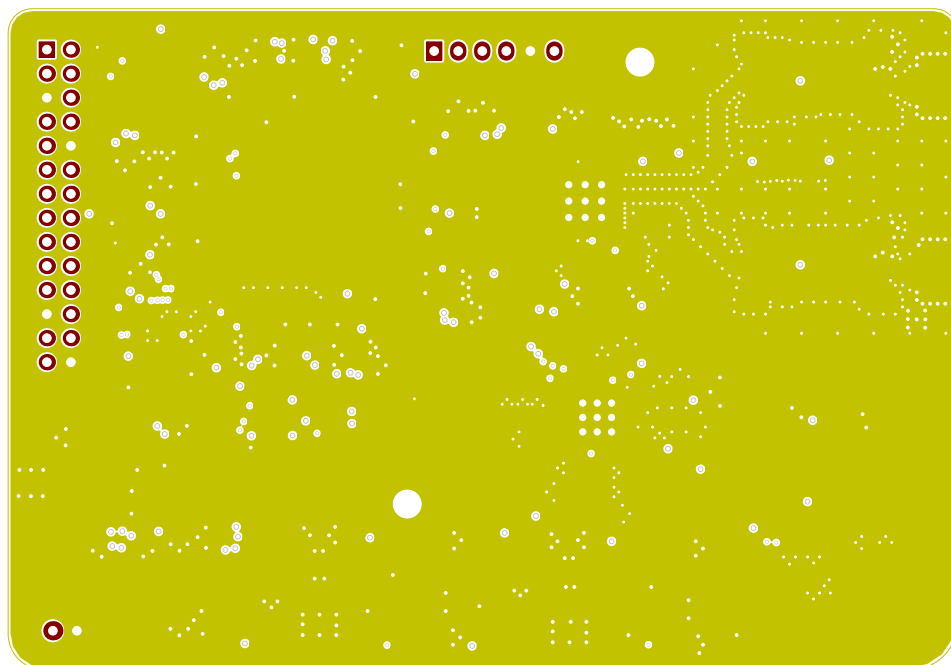


Figure F.9: Inner layer 1 Copper

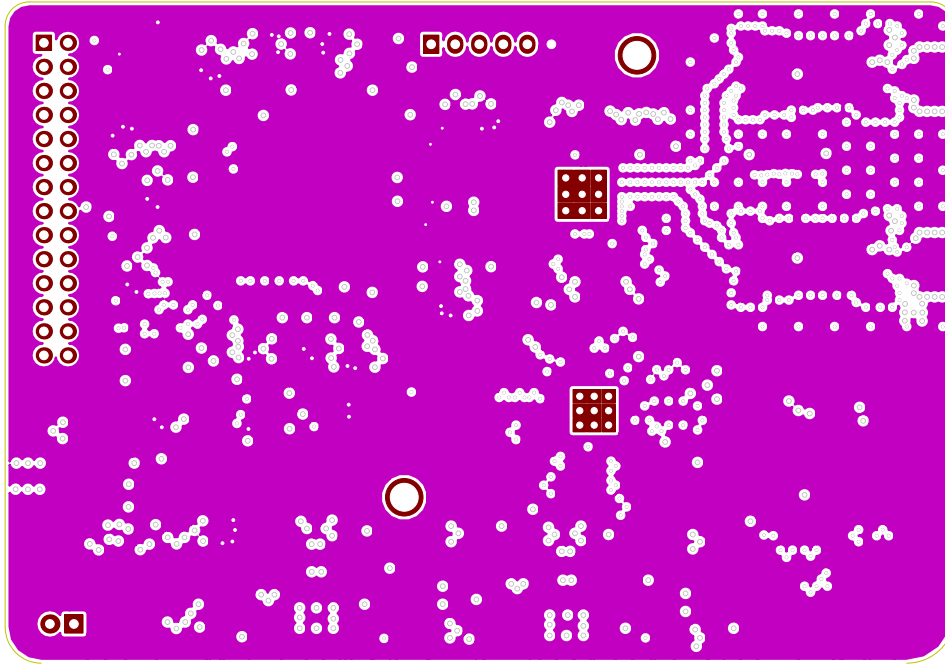


Figure F.10: Inner layer 2 Copper

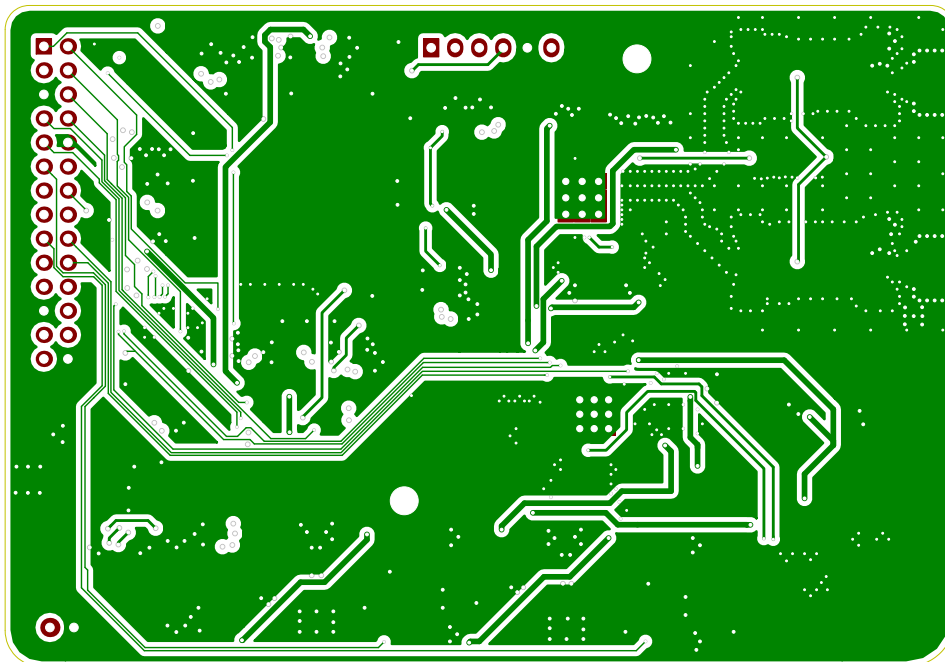


Figure F.11: Back Copper

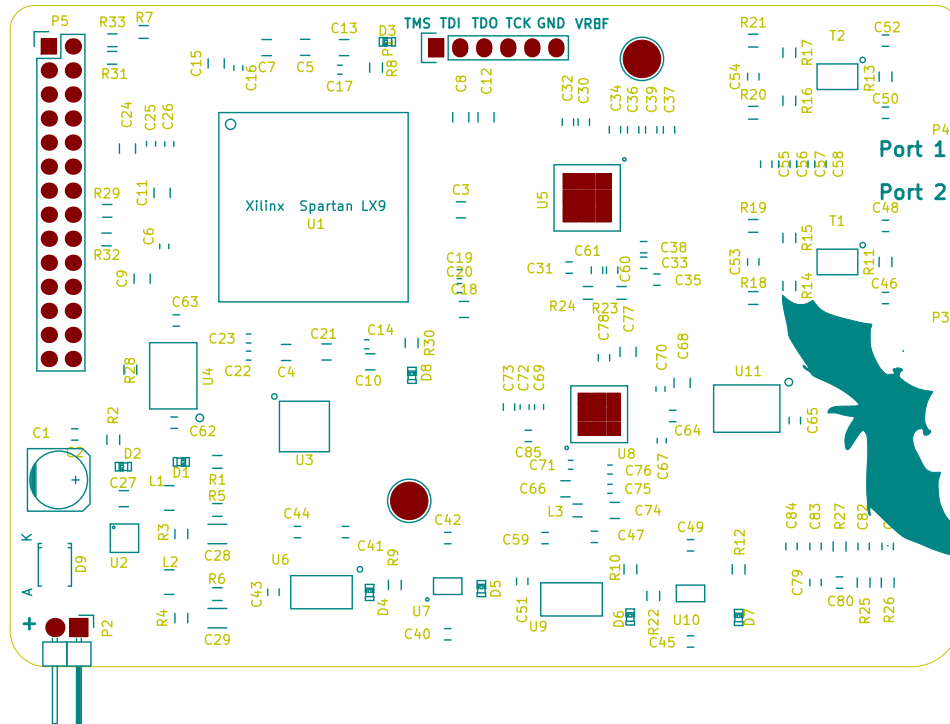


Figure F.12: Silk screen

Front-end filter bank

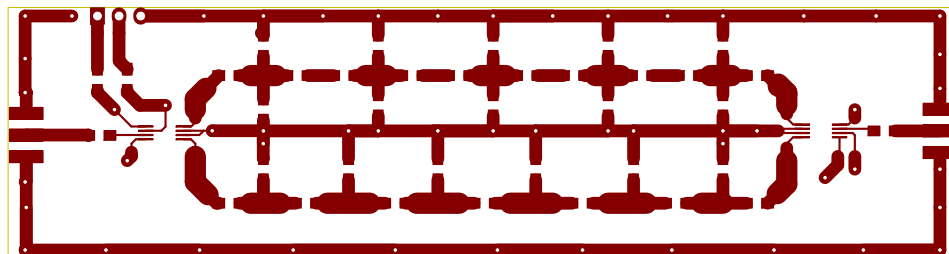


Figure F.13: Filterbank PCB Layout

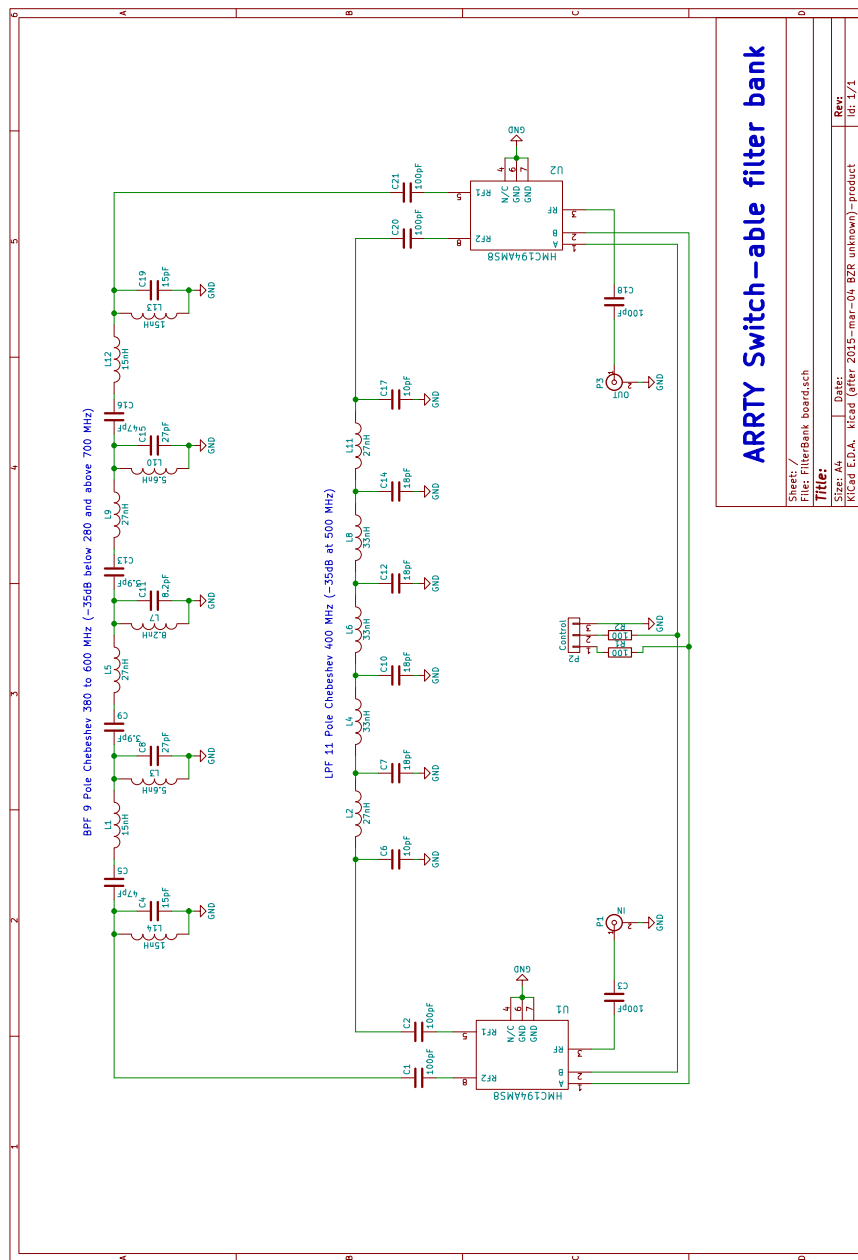


Figure F.14: Filterbank schematic

VHDL Code

The VHDL code that runs on the FPGA mainly functions as a FIFO memory to buffer the high-speed ADC data. Eight differential data lines along with a frame clock enter the ADC through 8-bit serial deserializers (SERDES). The framing clock is fed through a phase-lock loop to generate the internal clock

necessary to feed the SERDES units. 64 lines of data exit from the SERDES blocks at 125 MHz into the FIFO buffer. A state machine controls the FIFO buffer write and read clocks. A trigger mechanism starts the data capturing process until the requested sample length has been reached. While the trigger has not yet been initiated the number of stored elements in the buffer is kept at half of the sample length. This ensures that a trigger event will be captured in the centre of the sample stream. A four-wire serial interface allows the host to set the trigger level, configure the sample length, enable the trigger as well as download the sample stream after a successful capture.

Listing F.1: Main VHDL module for ARRTY FPGA

```

1  use IEEE.STD_LOGIC_1164.ALL;
2  use IEEE.STD_LOGIC_UNSIGNED.ALL;
3
4  use IEEE.NUMERIC_STD.ALL;
5
6  library UNISIM;
7  use UNISIM.VComponents.all;
8
9  entity Main is
10     Port (
11         -- Data from ADC
12         datain_p : in std_logic_vector(7 downto 0);
13         datain_n : in std_logic_vector(7 downto 0);
14         fclkin_p : in std_logic;
15         fclkin_n : in std_logic;
16         -- Clock from ADC
17         clk      : in std_logic;
18         din      : in std_logic;
19         dout     : out std_logic;
20         sen      : in std_logic;
21         LED      : out mstd_logic
22     );
23 end Main;
24
25 architecture Behavioral of Main is
26     --SERDES application module from Xilinx
27     component serdes_1_to_n_data_s8_diff generic (
28         -- Parameter to set the serdes factor 1..8
29         S : integer := 8;
30         -- Set the number of inputs and outputs
31         D : integer := 16;
32         -- Enable differential terminations
33         DIFF_TERM : boolean := TRUE
34     );
35     port (
36         use_phase_detector : in std_logic;
37         datain_p : in std_logic_vector(D-1 downto 0);
38         datain_n : in std_logic_vector(D-1 downto 0);
39         rxioclk      : in std_logic;
40         rxserdesstrobe : in std_logic;

```

```

41         reset          : in std_logic;
42         gclk            : in std_logic;
43         bitslip         : in std_logic;
44         debug_in : in std_logic_vector(1 downto 0);
45         data_out : out std_logic_vector((D*S)-1 downto 0);
46         debug : out std_logic_vector((2*D)+6 downto 0));
47     end component;
48
49     --SERDES application module from Xilinx
50     component serdes_1_to_n_clk_pll_s8_diff generic (
51         PLLD          : integer := 1;
52         CLKIN_PERIOD  : real := 10.0;
53         PLLX          : integer := 8;
54         S             : integer := 8;
55         BS            : boolean := TRUE;
56         DIFF_TERM     : boolean := TRUE
57     );
58     port (
59         clk_in_p      : in std_logic;
60         clk_in_n      : in std_logic;
61         reset         : in std_logic;
62         pattern1 : in std_logic_vector(S-1 downto 0);
63         pattern2 : in std_logic_vector(S-1 downto 0);
64         rx_ioclk      : out std_logic;
65         rx_serdesstrobe : out std_logic;
66         rx_bufg_pll_x1 : out std_logic;
67         bitslip       : out std_logic;
68         data_in       : out std_logic_vector(S-1 downto
69             0);
70         rx_bufpll_lckd : out std_logic);
71     end component;
72
73     component NBitComparator
74     generic (N:integer := 8);
75     port (
76         x : in STD_LOGIC_VECTOR(N-1 downto 0);
77         y : in STD_LOGIC_VECTOR(N-1 downto 0);
78         Gr : out STD_LOGIC;
79         Eq : out STD_LOGIC;
80         Lo : out STD_LOGIC);
81     end component;
82
83     -- Parameters for serdes factor and number of IO pins
84     -- Set the serdes factor to be maximum for diff lines
85     constant S : integer := 8;
86     -- Set the number of inputs
87     constant D : integer := 8 ;
88     -- Used for bus widths = serdes factor*number of inputs-1
89     constant DS : integer := (D*S)-1 ;
90
91     signal clk_iserdes_data : std_logic_vector(7 downto 0);
92     signal rx_bufg_x1 : std_logic;

```

APPENDIX F. AERIAL REAL-TIME TRANSIENT ANALYSER
DEVELOPMENT

180

```

92     signal rxd          : std_logic_vector(DS downto 0);
93     signal capture      : std_logic_vector(7 downto 0);
94     signal counter      : std_logic_vector(3 downto 0);
95     signal bitslip      : std_logic;
96     signal rst          : std_logic := '1';
97     signal rx_serdesstrobe : std_logic;
98     signal rx_bufpll_clk_xn : std_logic;
99     signal rx_bufpll_lckd  : std_logic;
100    signal not_bufpll_lckd  : std_logic;
101
102    --IPCore generated Block Ram
103    component FIFO is
104        port (
105            rst          : in std_logic;
106            wr_clk       : in std_logic;
107            rd_clk       : in std_logic;
108            din : in std_logic_vector(63 downto 0);
109            wr_en        : in std_logic;
110            rd_en        : in std_logic;
111            dout : out std_logic_vector(63 downto 0);
112            full         : out std_logic;
113            empty        : out std_logic;
114            wr_data_count: out std_logic_vector(12 downto 0);
115            rd_data_count: out std_logic_vector(12 downto 0)
116        );
117    end component;
118
119    signal wrDataCount: std_logic_vector(12 downto 0);
120    signal rdDataCount: std_logic_vector(12 downto 0);
121
122    component SerialInterface is
123        port (
124            clk          : in STD_LOGIC;
125            din          : in STD_LOGIC;
126            dout         : out STD_LOGIC;
127            sen          : in STD_LOGIC;
128            LED          : out STD_LOGIC;
129            fifo_clk     : out STD_LOGIC;
130            fifo_data : in STD_LOGIC_VECTOR(63 downto 0);
131            fifo_empty   : in STD_LOGIC;
132            fifo_reset   : out STD_LOGIC;
133            trigger      : out STD_LOGIC;
134            trigger_level : out STD_LOGIC_VECTOR(7 downto 0)
135        );
136        sample_length : out STD_LOGIC_VECTOR(15 downto 0);
137        capture_finished: in STD_LOGIC;
138        rst           : in STD_LOGIC
139    end component;
140
141    signal fifo_reset : std_logic;

```

APPENDIX F. AERIAL REAL-TIME TRANSIENT ANALYSER
DEVELOPMENT

181

```

142     signal fifo_data_out: std_logic_vector(63 downto 0);
143     signal trigger_level: std_logic_vector(7 downto 0);
144     signal sample_length: std_logic_vector(15 downto 0);
145     signal capture_complete: std_logic := '0';
146     signal samples_full : std_logic := '0';
147     signal samples_half : std_logic := '0';
148     signal fifo_empty   : std_logic := '0';
149     signal trigger      : std_logic := '0';
150     signal triggered    : std_logic := '0';
151     signal rd_clk       : std_logic := '0';
152     signal rd_clk_state_machine : std_logic := '0';
153     signal frameLocked  : std_logic := '0';
154     signal trig_seg1    : std_logic := '0';
155     signal trig_seg2    : std_logic := '0';
156     signal trig_seg3    : std_logic := '0';
157     signal trig_seg4    : std_logic := '0';
158     signal trig_seg5    : std_logic := '0';
159     signal trig_seg6    : std_logic := '0';
160     signal trig_seg7    : std_logic := '0';
161     signal trig_seg8    : std_logic := '0';
162
163 begin
164     --GENERAL-----
165     rst <= '0' ;
166
167     --FIFO Buffer control logic
168     comp_samples_full : NBitComparator
169         generic map (N => 13)
170         port map (
171             x => wrDataCount,
172             y => sample_length(15 downto 3),
173             Gr => open,
174             Eq => samples_full,
175             Lo => open);
176     comp_samples_half : NBitComparator
177         generic map (N => 13)
178         port map (
179             x => wrDataCount,
180             y => '0' & sample_length(15 downto 4),
181             Gr => samples_half,
182             Eq => open,
183             Lo => open);
184     comp_fifo_empty : NBitComparator
185         generic map (N => 13)
186         port map (
187             x => wrDataCount,
188             y => "0000000000000",
189             Gr => open,
190             Eq => fifo_empty,
191             Lo => open);
192     rd_clk <= (rx_bufg_x1 and samples_half and not triggered
               and not capture_complete);

```



```

193
194      --Evaluate trigger on the current 8 samples
195      comp_trig_seg1 : NBitComparator
196          generic map (N => 8)
197          port map (
198              x => rxd(7 downto 0),
199              y => trigger_level,
200              Gr => trig_seg1,
201              Eq => open,
202              Lo => open);
203      comp_trig_seg2 : NBitComparator
204          generic map (N => 8)
205          port map (
206              x => rxd(15 downto 8),
207              y => trigger_level,
208              Gr => trig_seg2,
209              Eq => open,
210              Lo => open);
211      comp_trig_seg3 : NBitComparator
212          generic map (N => 8)
213          port map (
214              x => rxd(23 downto 16),
215              y => trigger_level,
216              Gr => trig_seg3,
217              Eq => open,
218              Lo => open);
219      comp_trig_seg4 : NBitComparator
220          generic map (N => 8)
221          port map (
222              x => rxd(31 downto 24),
223              y => trigger_level,
224              Gr => trig_seg4,
225              Eq => open,
226              Lo => open);
227      comp_trig_seg5 : NBitComparator
228          generic map (N => 8)
229          port map (
230              x => rxd(39 downto 32),
231              y => trigger_level,
232              Gr => trig_seg5,
233              Eq => open,
234              Lo => open);
235      comp_trig_seg6 : NBitComparator
236          generic map (N => 8)
237          port map (
238              x => rxd(47 downto 40),
239              y => trigger_level,
240              Gr => trig_seg6,
241              Eq => open,
242              Lo => open);
243      comp_trig_seg7 : NBitComparator
244          generic map (N => 8)

```

```

245         port map (
246             x => rxd(55 downto 48),
247             y => trigger_level,
248             Gr => trig_seg7,
249             Eq => open,
250             Lo => open);
251 comp_trig_seg8 : NBitComparator
252     generic map (N => 8)
253     port map (
254         x => rxd(63 downto 56),
255         y => trigger_level,
256         Gr => trig_seg8,
257         Eq => open,
258         Lo => open);
259
260 triggered_latch : FDCE
261     generic map (INIT => '0')
262     port map (
263         Q => triggered,
264         C => trigger and (trig_seg1 or trig_seg2
                or trig_seg3 or trig_seg4 or
                trig_seg5 or trig_seg6 or trig_seg7 or
                trig_seg8),
265         CE => '1',
266         CLR => capture_complete,
267         D => '1');
268 --triggered <= trigger; Bypass trigger (uncomment)
269
270 --Bitslip frame locking process
271 bitslip_proc: process(bitslip, rx_bufpll_lckd)
272 begin
273     if(rising_edge(bitslip) and frameLocked = '0')then
274         if rxd = X"00000000000000FF" then
275             frameLocked <= '1';
276         end if;
277     end if;
278     if(rx_bufpll_lckd = '0')then
279         frameLocked <= '0';
280     end if;
281 end process;
282
283 --SERDES-----
284 --Clock Input, Generate ioclocks via PLL
285 clkin : serdes_1_to_n_clk_pll_s8_diff
286 generic map(
287     CLKIN_PERIOD => 8.0,
288     PLLD          => 1,
289     PLLX          => S,
290     S             => S,
291     BS            => TRUE)
292 port map (
293     clkin_p => fclkin_p,

```

APPENDIX F. AERIAL REAL-TIME TRANSIENT ANALYSER
DEVELOPMENT

184

```

294         clk_in_n => fclk_in_n ,
295         rx_ioclk => rx_bufpll_clk_xn ,
296         pattern1 => "00000000",
297         pattern2 => "00000001",
298         rx_serdes_strobe => rx_serdes_strobe ,
299         rx_bufg_pll_x1 => rx_bufg_x1 ,
300         bit_slip => bit_slip ,
301         reset => rst ,
302         data_in => clk_iserdes_data ,
303         rx_bufpll_lckd => rx_bufpll_lckd );
304
305     --8 Data Inputs being deserialised
306     not_bufpll_lckd <= not rx_bufpll_lckd;
307
308     data_in : serdes_1_to_n_data_s8_diff generic map(
309         S => S,
310         D => D,
311         DIFF_TERM => TRUE
312     )
313     port map (
314         use_phase_detector => '1',
315         data_in_p => data_in_p ,
316         data_in_n => data_in_n ,
317         rx_ioclk => rx_bufpll_clk_xn ,
318         rx_serdes_strobe => rx_serdes_strobe ,
319         gclk => rx_bufg_x1 ,
320         bit_slip => bit_slip and not frameLocked ,
321         reset => not_bufpll_lckd ,
322         debug_in => "00" ,
323         data_out => rxd ,
324         debug => open );
325
326
327     --RAMBLOCK-----
328     fifoMemory : FIFO
329     port map (
330         rst => fifo_reset ,
331         wr_clk => rx_bufg_x1 ,
332         rd_clk => rd_clk or rd_clk_state_machine ,
333         din => rxd ,
334         wr_en => trigger and not capture_complete ,
335         rd_en => '1' ,
336         dout => fifo_data_out ,
337         full => open ,
338         empty => open ,
339         wr_data_count => wrDataCount ,
340         rd_data_count => rdDataCount
341     );
342
343     --STATE MACHINE-----
344     stateMachine : SerialInterface
345     port map (

```

```

346         clk => clk ,
347         din  => din ,
348         dout => dout ,
349         sen  => sen ,
350         LED  => LED,
351         fifo_clk  => rd_clk_state_machine ,
352         fifo_data => fifo_data_out ,
353         fifo_empty => fifo_empty ,
354         fifo_reset => fifo_reset ,
355         trigger   => trigger ,
356         trigger_level => trigger_level ,
357         sample_length => sample_length ,
358         capture_finished => capture_complete ,
359         rst        => not_bufpll_lckd
360     );
361
362     — Capture latch , notifies system that the capture is
       complete
363     capture_complete_latch : FDPE
364     generic map (
365         INIT => '0')
366     port map (
367         Q  => capture_complete ,
368         C  => rx_bufg_x1 and fifo_empty ,
369         CE => '1' ,
370         PRE => samples_full and triggered ,
371         D  => '0');
372 end Behavioral;
```

Listing F.2: Comparator VHDL module for ARRTY FPGA

```

373 use IEEE.STD_LOGIC_1164.ALL;
374 use IEEE.STD_LOGIC_UNSIGNED.ALL;
375
376 entity NBitComparator is
377     generic (N:integer := 8);
378     port (
379         x  : in  STD_LOGIC_VECTOR(N-1 downto 0);
380         y  : in  STD_LOGIC_VECTOR(N-1 downto 0);
381         Gr : out STD_LOGIC;
382         Eq : out STD_LOGIC;
383         Lo : out STD_LOGIC);
384 end NBitComparator;
385
386 architecture NBitComparator of NBitComparator is
387 begin
388     process(x, y)
389     begin
390         Gr <= '0';
391         Eq <= '0';
392         Lo <= '0';
393         if (x > y) then
394             Gr <= '1';
```

```

395         elsif(x = y) then
396             Eq <= '1';
397         elsif (x < y) then
398             Lo <= '1';
399         end if;
400     end process;
401 end NBitComparator;

```

Listing F.3: Serial state machine VHDL module for ARRTY FPGA

```

402 use IEEE.STD_LOGIC_1164.ALL;
403 use IEEE.NUMERIC_STD.ALL;
404 library UNISIM;
405 use UNISIM.VComponents.all;
406
407 entity SerialInterface is
408     Port (
409         --Interface connections
410         clk      : in  STD_LOGIC;
411         din      : in  STD_LOGIC;
412         dout     : out STD_LOGIC;
413         sen      : in  STD_LOGIC;
414         LED      : out STD_LOGIC;
415
416         --Internal control connections
417         fifo_clk      : out STD_LOGIC;
418         fifo_data     : in  STD_LOGIC_VECTOR(63 downto 0);
419         fifo_empty     : in  STD_LOGIC;
420         fifo_reset     : out STD_LOGIC;
421         trigger        : out STD_LOGIC;
422         trigger_level  : out STD_LOGIC_VECTOR(7 downto 0)
423         ;
424         sample_length  : out STD_LOGIC_VECTOR(15 downto
425         0);
426         capture_finished : in  STD_LOGIC;
427         rst             : in  STD_LOGIC
428     );
429 end SerialInterface;
430
431 architecture Behavioral of SerialInterface is
432     --Setup state machine variables
433     type state_type is (idle, read_data, set_triglevel,
434         set_samplelength, trigger_adc);
435     signal present_state : state_type := idle;
436     signal next_state    : state_type := idle;
437
438     --State machine functionality storage registers
439     signal trigger_level_storage : std_logic_vector(7 downto
440     0) := "00000000";
441     signal sample_length_storage : std_logic_vector(15 downto
442     0) := "0000010000000000";
443
444     --State machine command register

```

```

440     component shift_register
441         port (
442             d    : in  STD_LOGIC;
443             clk  : in  STD_LOGIC;
444             q    : out STD_LOGIC_VECTOR (15 downto 0)
445         );
446     end component;
447
448     signal reg_out    : std_logic_vector(15 downto 0);
449
450     signal triggerTemp : std_logic;
451     signal trigger_to_latch : std_logic;
452
453     signal dout_buffer : std_logic := '0';
454     signal din_buffer  : std_logic;
455     signal clk_buffer  : std_logic;
456     signal sen_buffer  : std_logic;
457
458     signal fifo_clk_temp : std_logic := '0';
459 begin
460     sync_proc: process(clk_buffer, rst)
461         --readout command variables
462     begin
463         if (rst = '1') then
464             present_state <= idle;
465         elsif rising_edge(clk_buffer) then
466             present_state <= next_state;
467         end if;
468     end process sync_proc;
469
470     comb_proc: process(next_state, sen_buffer)
471         --readout command variables
472     begin
473         --Handle present state
474         if (falling_edge(sen_buffer)) then
475             case present_state is
476             when idle =>
477                 trigger_to_latch <= '0';
478                 case reg_out is
479                     when "0000000000000000" =>
480                         next_state <= idle;
481                     when "0000000000000001" =>
482                         next_state <=
483                             set_triglevel;
484                     when "0000000000000010" =>
485                         next_state <=
486                             set_samplelength;
487                     when "0000000000000011" =>
488                         trigger_to_latch <= '1';
489                     when others =>
490                         next_state <= idle;
491                 end case;

```

```

490         when set_triglevel =>
491             trigger_level_storage <= reg_out(7 downto
492                 0);
493             next_state <= idle;
494         when set_samplelength =>
495             sample_length_storage <= reg_out(15 downto
496                 0);
497             next_state <= idle;
498         when others =>
499             next_state <= idle;
500         end case;
501     end if;
502 end process comb_proc;
503 trigger_level <= trigger_level_storage;
504 sample_length <= sample_length_storage;
505 LED <= capture_finished;
506 trigger <= triggerTemp;
507
508 —Process to handle the fifo clocking
509 fifo_read_proc: process(clk_buffer)
510     variable fifo_pointer      : integer := 0;
511     variable fifo_jump_pointer : integer := 0;
512     variable divider           : integer := 0;
513     variable div_counter       : integer := 0;
514 begin
515     if rising_edge(clk_buffer) then
516         if capture_finished = '1' then
517             fifo_jump_pointer := 8*(
518                 fifo_pointer mod 8) + divider;
519             dout_buffer <= fifo_data(
520                 fifo_jump_pointer);
521             fifo_pointer := fifo_pointer + 1;
522
523             div_counter := div_counter + 1;
524             if div_counter = 8 then
525                 div_counter := 0;
526                 divider := divider + 1;
527             end if;
528
529             if fifo_pointer = 64 then
530                 fifo_pointer := 0;
531                 divider := 0;
532                 fifo_clk_temp <= '1';
533             else
534                 fifo_clk_temp <= '0';
535             end if;
536         else
537             fifo_pointer := 0;
538             fifo_jump_pointer := 0;
539             divider := 0;
540             div_counter := 0;
541         end if;
542     end if;

```

APPENDIX F. AERIAL REAL-TIME TRANSIENT ANALYSER
DEVELOPMENT

189

```

538         end if;
539     end process fifo_read_proc;
540     fifo_clk <= fifo_clk_temp;
541
542     --Serial logic elements
543     serial_shiftreg : shift_register
544         port map (
545             d    => din_buffer ,
546             clk => clk_buffer and sen_buffer ,
547             q    => reg_out
548         );
549
550     trigger_latch : FDCE
551         generic map (
552             INIT => '0') -- Initial value of register
553                        ('0' or '1')
554         port map (
555             Q    => triggerTemp ,
556             C    => trigger_to_latch ,
557             CE   => '1' ,
558             CLR  => sen_buffer ,
559             D    => '1');
560
561     fifo_reset <= fifo_empty and sen_buffer;
562
563     --Input and output buffers for outside world
564     dout_buf : OBUF
565         generic map (
566             DRIVE => 12,
567             SLEW  => "SLOW")
568         port map (
569             O => dout ,
570             I => (sen_buffer and not capture_finished)
571                  or (not sen_buffer and
572                     capture_finished) or (capture_finished
573                     and dout_buffer));
574
575     din_buf : IBUF
576         generic map (
577             IBUFLOW_PWR => FALSE)
578         port map (
579             O => din_buffer ,
580             I => din);
581
582     clk_buf : IBUF
583         generic map (
584             IBUFLOW_PWR => FALSE)
585         port map (

```



```

586             O => sen_buffer ,
587             I => sen));
588 end Behavioral;

```

Listing F.4: Shift Register VHDL module for ARRTY FPGA

```

589 use IEEE.STD_LOGIC_1164.ALL;
590 use IEEE.STD_LOGIC_UNSIGNED.ALL;
591
592 entity shift_register is
593     Port ( d      : in  STD_LOGIC;
594           clk     : in  STD_LOGIC;
595           q       : out STD_LOGIC_VECTOR (15 downto 0));
596 end shift_register;
597
598 architecture Behavioral of shift_register is
599     signal shift_reg : STD_LOGIC_VECTOR(15 downto 0) := "
600         0000000000000000";
601
602     -- shift register
603     process (clk)
604     begin
605         if (rising_edge(clk)) then
606             shift_reg(14 downto 0) <= shift_reg(15 downto 1);
607             shift_reg(15) <= D;
608         end if;
609     end process;
610
611     q <= shift_reg;
612 end Behavioral;

```

Host Python Code

The host code manages all of the subsystems that form part of the analyser. Objects are created for the clock, ADC and FPGA. At this point, the clock is configured for 1 GHz while the ADC is set to output a framing signal. The framing data enables the FPGA to appropriately align its data onto the FIFO input bus. After this, the ADC is set to sample on port one with all four of its ADC's interleaved at their highest programmable gain. Making use of the serial interface to the FPGA, the sample length and trigger level are configured. With the capture configured, the trigger is set to start capturing a stream of data. At this point, the host listens for the capture complete signal whereafter it downloads the data off of the device.

Listing F.5: Main Python module for ARRTY host

```

556 import ADCCon
557 import SynthCon

```

APPENDIX F. AERIAL REAL-TIME TRANSIENT ANALYSER
DEVELOPMENT

191

```

558 import FPGACon
559
560 import sys
561 import time
562 import numpy as np
563 import matplotlib.pyplot as plt
564
565 class ARRTY():
566     def __init__(self):
567         #Turn on Synth
568         self.synth = SynthCon.SynthCon()
569         self.synth.power(True)
570         time.sleep(0.01)
571         self.synth.setFrequency(1000.001e6)
572
573         #Turn on ADC
574         self.adc = ADCCon.ADCCon()
575         time.sleep(0.01)
576
577         time.sleep(1)
578         #Initialise FPGA
579         self.fpga = FPGACon.FPGACon()
580         time.sleep(0.01)
581
582         #Set ramping pattern
583         self.adc.setFramingPattern(False)
584         #self.adc.setTestValuePattern(200)
585         time.sleep(0.1)
586
587         #Set ADC internal gain to 50x
588         self.adc.setGain()
589
590         triggerLevel = int(sys.argv[1])
591         print("Setting_trigger_level_to_"+str(triggerLevel)
592             +"..."),
593         self.fpga.setTriggerLevel(triggerLevel)
594         print("[DONE]")
595         sampleLength = int(sys.argv[2])
596         print("Setting_sample_length_to_"+str(sampleLength)
597             +"..."),
598         self.fpga.setSampleLength(sampleLength)
599         print("[DONE]")
600
601     def shutdown(self):
602         self.adc.power(False)
603         self.synth.power(False)
604
605 try:
606     arrty = ARRTY()
607
608     while(1):
609         arrty.fpga.setTrigger()

```

```

608         print("Waiting for sample ..."),
609         while( arrty.fpga.readDOUT() == 0):
610             time.sleep(0.001)
611         print("[CAPTURED]")
612         print("Reading sample:")
613         sample = arrty.fpga.readData()
614
615     except KeyboardInterrupt:
616         arrty.shutdown()
617         print("Putting ARRTY to sleep ...")
618         GPIO.cleanup()

```

Listing F.6: ADC Control Python module for ARRTY host

```

1  import RPi.GPIO as GPIO
2  import time
3  import numpy as np
4
5  '''Pin connections'''
6  PWR_EN = 2
7  CSN = 3
8  SDATA = 4
9  SCLK = 17
10 RESETN = 27
11 PWRDWN = 22
12
13 class ADCCon():
14     def __init__(self):
15         '''Setup pin configuration'''
16         self.setupPins()
17
18         self.power(False)
19         time.sleep(0.001)
20
21         print("#####Initiating ADC#####")
22         print("Applying power to ADC")
23         self.power(True)
24
25         print("Resetting ADC")
26         self.reset()
27         print("Setting the ADC into power down")
28         self.powerDown(True)
29
30         print("Configuring ADC"),
31         #Modes of operation and clock divide factor
32         reg31 = 0b0000000000000000;
33         reg31 |= (1<<0) # Set single channel mode
34         reg31 |= (0<<8) # Set clock div to 1
35         self.writeReg(0x31, reg31)
36
37         #Input Select
38         reg3A = 0b0000000000000000;
39         reg3B = 0b0000000000000000;

```

APPENDIX F. AERIAL REAL-TIME TRANSIENT ANALYSER
DEVELOPMENT

193

```

40      #Connect all 4 ADC's to a single port 1
41      reg3A |= (1<<1)
42      reg3A |= (1<<9)
43      reg3B |= (1<<1)
44      reg3B |= (1<<9)
45      self.writeReg(0x3A, reg3A)
46      self.writeReg(0x3B, reg3B)
47
48      #Full-scale control (Default)
49      #Current control (Default)
50      #External Common Mode Voltage Buffer Driving
51      Strength (Default)
52      reg50 = 0b0000000000000000;
53      reg50 |= (0b10<<4)
54      self.writeReg(0x50, reg50)
55
56      #Start-up and Clock Jitter Control
57      reg56 = 0b0000000000000000;
58      reg30 = 0b0000000000000000;
59      reg56 |= (0b011<<0) # Set start up time to 13 -
60      17.3us
61      reg30 |= (1<<0) # Clock jitter at 160 fsrms (
62      Lowest power)
63      self.writeReg(0x56, reg56)
64      self.writeReg(0x30, reg30)
65
66      #LVDS Output Configuration and Control
67      #Low clock frequency, advance and delay all
68      inactive
69      #LCLK phase shifted 90 degrees
70      #LSB first
71      #Straight offset binary
72
73      #LVDS Drive Strength
74      #3.5mA drive (Standard)
75
76      #LVDS Internal Termination Programmability
77      #Disabled
78
79      #Power mode control
80      #Inactive
81
82      #Programmable gain
83      #1x gain
84      #Branch gains - 0dB
85
86      #Analog input invert
87      #Inactive
88
89      #LVDS Test Patterns
90      self.setFramingPattern(True)

```

```

88         time.sleep(0.001)
89         print("Setting the ADC into power up mode")
90         self.powerDown(False)
91
92         print("[DONE]")
93
94     def setGain(self, gain = 0b1101):
95         '''Set the programmable gain of the adc in dB
96         .....0b1101 - 50x'''
97         #Enable course gain
98         reg33 = 0b0000000000000000;
99         reg33 |= (0b1<<0)
100        self.writeReg(0x33, reg33)
101
102        #Set gain for single channel setup
103        reg2B = 0b0000000000000000;
104        reg2B |= (gain<<8)
105        self.writeReg(0x2B, reg2B)
106
107    def setTestValuePattern(self, value):
108        '''Turn on a test pattern'''
109        reg25 = 0b0000000000000000;
110        reg26 = 0b0000000000000000;
111        #Enable single code
112        reg25 |= (0b001<<4)
113        reg26 |= (value<<8)
114        self.writeReg(0x25, reg25)
115        self.writeReg(0x26, reg26)
116
117
118    def setFramingPattern(self, enabled):
119        '''Turn on a framing pattern'''
120        reg25 = 0b0000000000000000;
121        reg26 = 0b0000000000000000;
122        #Enable single code
123        if enabled:
124            reg25 |= (0b001<<4)
125        else:
126            reg25 |= (0b000<<4)
127        reg26 |= (0b00000001<<8)
128        self.writeReg(0x25, reg25)
129        self.writeReg(0x26, reg26)
130
131    def setRampPattern(self, enabled):
132        '''Turn on a ramping pattern'''
133        reg25 = 0b0000000000000000;
134        #Enable single code
135        if enabled:
136            reg25 |= (0b100<<4)
137        else:
138            reg25 |= (0b000<<4)
139        self.writeReg(0x25, reg25)

```

```

140
141     def setupPins(self):
142         '''Configures the behavior of the GPIO pins'''
143         GPIO.setwarnings(False)
144         GPIO.setmode(GPIO.BCM) # Sets numbering scheme
145         GPIO.setup(PWREN, GPIO.OUT)
146         GPIO.setup(RESETN, GPIO.OUT)
147         GPIO.setup(PWRDWN, GPIO.OUT)
148         GPIO.setup(CSN, GPIO.OUT)
149         GPIO.setup(SDATA, GPIO.OUT)
150         GPIO.setup(SCLK, GPIO.OUT)
151
152         '''Set default states'''
153         GPIO.output(PWRDWN, GPIO.HIGH)
154         GPIO.output(PWRDWN, GPIO.HIGH)
155         GPIO.output(RESETN, GPIO.HIGH)
156         GPIO.output(PWREN, GPIO.LOW)
157
158     def power(self, powerOn=True):
159         '''Turns on ARRTY regulators to ADC module'''
160         if powerOn:
161             GPIO.output(PWREN, GPIO.HIGH)
162         else:
163             GPIO.output(PWREN, GPIO.LOW)
164
165     def reset(self):
166         '''Applies a reset signal to the module'''
167         GPIO.output(RESETN, GPIO.LOW)
168         time.sleep(0.001)
169         GPIO.output(RESETN, GPIO.HIGH)
170
171     def powerDown(self, state=True):
172         '''Sets the module into power down mode'''
173         if state:
174             GPIO.output(PWRDWN, GPIO.HIGH)
175         else:
176             GPIO.output(PWRDWN, GPIO.LOW)
177
178     def writeReg(self, addr, data):
179         '''Writes a register with the specified data at
180         .....an given address'''
181
182         '''Bring Clock low'''
183         GPIO.output(SCLK, GPIO.LOW)
184         '''Bring CSN low to initiate write'''
185         GPIO.output(CSN, GPIO.LOW)
186
187         '''Clock in 8 bits of address data'''
188         for i in np.arange(8):
189             '''Set bit'''
190             if ((1<(7-i))&addr) > 0:
191                 GPIO.output(SDATA, GPIO.HIGH)

```

```

192         else:
193             GPIO.output(SDATA, GPIO.LOW)
194             '''Clock_bit'''
195             GPIO.output(SCLK, GPIO.HIGH)
196             GPIO.output(SCLK, GPIO.LOW)
197
198         '''Clock_in_16_bits_of_payload_data'''
199         for i in np.arange(16):
200             '''Set_bit'''
201             if ((1<<(15-i))&data) > 0:
202                 GPIO.output(SDATA, GPIO.HIGH)
203             else:
204                 GPIO.output(SDATA, GPIO.LOW)
205             '''Clock_bit'''
206             GPIO.output(SCLK, GPIO.HIGH)
207             GPIO.output(SCLK, GPIO.LOW)
208
209         '''Finish_write_by_bringing_CSN_high_again'''
210         GPIO.output(CSN, GPIO.HIGH)

```

Listing F.7: Synth Control Python module for ARRTY host

```

214 import RPi.GPIO as GPIO
215 import time
216 import numpy as np
217 from fractions import Fraction
218
219 '''Pin_connections'''
220 PWR_EN = 14
221 CEN = 15
222 SEN = 18
223 SDO = 23
224 SCK = 24
225 SDI = 10
226
227 '''Verbose'''
228 verbose = True
229
230 class SynthCon():
231     def __init__(self):
232         '''Set_up_the_pins_to_the_digital_board'''
233         self.setupPins()
234
235         self.power(False)
236         time.sleep(0.001)
237
238     def setFrequency(self, freq):
239         '''Sets_the_registers_to_the_desired_frequency'''
240         ref = 50e6
241         if verbose:
242             print("Calculating_registers_for:" + str(
243                 freq) + " Hz")
244         '''Calculate_NOUT'''

```

```

244      '''VCO_runs_from_1.5_GHz_up_to_3_GHz, NOUT_sets_
        the_VCO_output_divider'''
245      NOUT = 2*np.floor(3e9/(2*freq))
246
247      '''Calculate_the_reference_divider'''
248      '''PFD_needs_to_run_below_70_MHz'''
249      RDIV = np.ceil(ref/70e6)
250      FPDF = ref/RDIV
251
252      '''Calculating_NDIV'''
253      ratio = freq/ref
254      if(NOUT == 0):
255          NDIV = ratio*RDIV
256      else:
257          NDIV = ratio*RDIV*NOUT
258      NINT = np.floor(NDIV)
259      NFRAC = np.ceil((2**24)*(NDIV-NINT))
260
261      '''Calculate_the_exact_frequency_mode_register'''
262      fraction = Fraction(float(NDIV)-float(NINT))
263      EXACT = 1/(fraction.limit_denominator())
264
265      '''Calculate_charge_pump_settings'''
266      if(NOUT == 0):
267          FVCO = freq
268      else:
269          FVCO = freq*NOUT
270      ICP = 1.1e-3 + (2.3e-3 - 1.1e-3)*(FVCO-1.5e9)/(3e9
        -1.5e9)
271      ICPOffset = ((2.5e-9) + 4*(1/FVCO))*(ICP/(1/(FPFD)
        ))
272
273      '''Lock_detection_window_settings'''
274      LDWindow = (((ICPOffset)/(FPFD*ICP)))+(2.66e-9)+(1/
        FPFD))/2
275
276      '''Print_out_settings'''
277      if verbose and False:
278          print("-----PLL Configuration-----")
279          print("Freq____: "+str(freq*1e-9)+"_GHz")
280          print("Ref____: "+str(ref*1e-6)+"_MHz")
281          print("Ratio___: "+str(ratio))
282          print("NOUT____: "+str(NOUT))
283          print("RDIV____: "+str(RDIV))
284          print("FPFD____: "+str(FPDF*1e-6)+"_MHz")
285          print("NDIV____: "+str(NDIV))
286          print("NINT____: "+str(NINT))
287          print("NFRAC___: "+str(NFRAC))
288          print("DEN____: "+str(EXACT))
289          print("FVCO____: "+str(FVCO*1e-9)+"_GHz")
290          print("ICP____: "+str(ICP*1e6)+"_uA")

```



```

291         print("ICPOff: "+str(ICPOffset*1e6)+" uA"
292               )
293         print("LD_Wind: "+str(LDWindow*1e9)+" ns")
294         print("_____")
295         ''' Check_minimum_divide_ratio (20_for_fractional_
296             mode) '''
297         if (NINT < 20):
298             print("ERROR: Minimum_divide_ratio_not_met
299                 ")
300             return
301
302         if verbose:
303             print(" Configuring_device ..."),
304
305         ''' Configure_reference_divider '''
306         RDIV = int(RDIV)
307         self.writeReg(0x02,RDIV)
308
309         ''' Configure_delta-sigma_modulator '''
310         DCM = 2
311         DCM = DCM|2<<2
312         DCM = DCM|4<<4
313         DCM = DCM|0<<7
314         DCM = DCM|1<<8
315         DCM = DCM|1<<9
316         DCM = DCM|1<<10
317         DCM = DCM|1<<11
318         self.writeReg(0x06,DCM)
319
320         ''' Configure_autocal '''
321         AC = 0x2046
322         self.writeReg(0x0A,AC)
323
324         ''' Configure_charge_pump_current '''
325         ''' Calculate_register '''
326         #CPC = 0x30E5CB
327         CPC = 1<<21 # Enable up offset
328         CP = int(np.round(ICP/20e-6))
329         CPC = CPC|CP<<0
330         CPC = CPC|CP<<7
331         CPO = int(np.round(ICPOffset/5e-6))
332         CPC = CPC|CPO<<14
333         self.writeReg(0x09,CPC)
334
335         ''' Configure_lock_detect '''
336         #LD = 0x94D
337         LD = 5<<0
338         LD = LD|1<<3
339         LD = LD|1<<6
340         divide,speed = self.LDLookup(LDWindow)
341         LD = LD|int(divide)<<7
342         LD = LD|int(speed)<<10

```

```

340         self.writeReg(0x07,LD)
341
342         ''' Configure VCO subsystem, VSPI clock = 12.5MHz
           '''
343         #self.writeReg(0x05,0xFF88) # Enable output stage
344         switchSettings = self.readReg(0x10)
345         switchSettings = switchSettings&0b01111111
346         self.writeReg(0x05,0xF98)
347         # Low phase noise mode
348         # Enable RF_N and RF_P
349         # Internal termination
350         time.sleep(0.001)
351         self.writeReg(0x05,0x4B38) #Set output gain to 1.4
           V
352         time.sleep(0.001)
353         #self.writeReg(0x01,0xFF88) #Set output on
354         VCOSS = (0x0|switchSettings<<1)<<7
355         self.writeReg(0x05,VCOSS)
356         time.sleep(0.001)
357
358         ''' Programming the frequency of operation '''
359         self.writeReg(0x03,int(NINT))
360         #self.writeReg(0x0C,int(EXACT))
361         self.writeReg(0x04,long(NFRAC))
362
363         ''' Setting the output divider '''
364         switchSettings = self.readReg(0x10)
365         switchSettings = switchSettings&0b01111111
366         NOUTReg = 0<<0
367         NOUTReg = NOUTReg|2<<3
368         NOUTReg = NOUTReg|int(NOUT)<<7
369         self.writeReg(0x05,NOUTReg) #0x90 Set VCO div
370         time.sleep(0.001)
371         VCOSS = (0x0|switchSettings<<1)<<7
372         self.writeReg(0x05,VCOSS)
373         time.sleep(0.001)
374
375         print(" [DONE] ")
376         print(" Verifying configuration ... "),
377         if verbose and False:
378             print(" 0x02: " + hex(RDIV))
379             print(" 0x03: " + hex(NINT))
380             print(" 0x04: " + hex(NFRAC))
381             print(" 0x05: " + hex(VCOSS))
382             print(" 0x06: " + hex(DCM))
383             print(" 0x07: " + hex(self.readReg(0x07)))
384             print(" 0x08: " + hex(self.readReg(0x08)))
385             print(" 0x09: " + hex(CPC))
386             print(" 0x0A: " + hex(self.readReg(0x0A)))
387             print(" 0x0B: " + hex(self.readReg(0x0B)))
388             print(" 0x0C: " + hex(self.readReg(0x0C)))
389         if(RDIV == self.readReg(0x02) and

```

```

390         DCM = self.readReg(0x06) and
391         CPC = self.readReg(0x09) and
392         LD = self.readReg(0x07) and
393         VCOSS = self.readReg(0x05) and
394         NINT = self.readReg(0x03) and
395         NFRAC = self.readReg(0x04)):
396             print("[SUCCESS]")
397     else:
398         print("[Fail]")
399
400     time.sleep(0.001)
401     return self.checkLock()
402
403
404
405
406     def checkLock(self):
407         '''Polls the lock register'''
408         if verbose:
409             print("Checking Lock..."),
410         GPO2Reg = self.readReg(0x12)
411         locked = GPO2Reg & 0b10
412         if locked > 0:
413             if verbose:
414                 print("[LOCKED]")
415             return True
416         else:
417             if verbose:
418                 print("[NOT LOCKED]")
419             return False
420
421
422     def readVCOTuneRegister(self):
423         '''Reads the status of the VCO tune register'''
424         if verbose:
425             print("VCO Subsystem Status:")
426         VCOTune = self.readReg(0x10)
427         VCOSS = VCOTune & 0b01111111
428         if VCOTune & 0b10000000 > 0:
429             autoCal = 'Busy'
430         else:
431             autoCal = 'Not Busy'
432         if verbose:
433             print("Current VCO setting: " + str(VCOSS))
434             print("AutoCal: " + str(autoCal))
435
436     def setupPins(self):
437         '''Sets up the GPIO pins'''
438         '''Configure the behaviour of the GPIO pins'''
439         GPIO.setwarnings(False)
440         GPIO.setmode(GPIO.BCM)
441         GPIO.setup(PWREN, GPIO.OUT)

```

```

442         GPIO.setup(CEN, GPIO.OUT)
443         GPIO.setup(SEN, GPIO.OUT)
444         GPIO.setup(SDO, GPIO.IN)
445         GPIO.setup(SCK, GPIO.OUT)
446         GPIO.setup(SDI, GPIO.OUT)
447
448         '''Set_default_states'''
449         GPIO.output(PWREN, GPIO.LOW)
450         GPIO.output(CEN, GPIO.LOW)
451         GPIO.output(SEN, GPIO.LOW)
452         GPIO.output(SDI, GPIO.LOW)
453         GPIO.output(SCK, GPIO.LOW)
454
455
456     def power(self, powerOn=True):
457         '''Turn_on_power_to_the_module'''
458         if powerOn:
459             if verbose:
460                 print("Applying_power_to_Synth")
461             GPIO.output(PWREN, GPIO.HIGH)
462             time.sleep(0.001)
463
464             if verbose:
465                 print("Setting_up_SPI_mode_to_HMC")
466             self.setSPIMode("HMC")
467             time.sleep(0.001)
468
469             '''Configure_the_function_of_the_SDO/LD_line'''
470             self.writeReg(0x0F, (0b000<<5)|1)
471             time.sleep(0.001)
472
473             if verbose:
474                 '''Check_the_device_ID'''
475                 self.checkDeviceID()
476         else:
477             if verbose:
478                 print("Pulling_serial_pins_low")
479             GPIO.output(SCK, GPIO.LOW)
480             GPIO.output(SDI, GPIO.LOW)
481             time.sleep(0.001)
482             if verbose:
483                 print("Disabling_serial_port")
484             GPIO.output(SEN, GPIO.LOW)
485             time.sleep(0.001)
486             if verbose:
487                 print("Disabling_device")
488             GPIO.output(CEN, GPIO.LOW)
489             time.sleep(0.001)
490             if verbose:
491                 print("Shutting_down_power")

```

```

492         GPIO.output(PWREN, GPIO.LOW)
493
494     def setSPIMode(self, mode="HMC"):
495         '''Sets up the communication mode to the module,
496         must
497         .....be the first command sent to the module.'''
498         GPIO.output(SCK, GPIO.LOW)
499         GPIO.output(SEN, GPIO.LOW)
500         GPIO.output(CEN, GPIO.LOW)
501         time.sleep(0.001)
502         GPIO.output(CEN, GPIO.HIGH)
503         time.sleep(0.001)
504
505         if mode == "HMC":
506             if verbose:
507                 print("Pulsing SEN")
508             GPIO.output(SEN, GPIO.LOW)
509             time.sleep(0.001)
510             GPIO.output(SEN, GPIO.HIGH)
511             time.sleep(0.001)
512             GPIO.output(SEN, GPIO.LOW)
513         elif mode == "Open":
514             GPIO.output(SCK, GPIO.LOW)
515             time.sleep(0.001)
516             GPIO.output(SCK, GPIO.HIGH)
517             time.sleep(0.001)
518             GPIO.output(SCK, GPIO.LOW)
519         else:
520             print("This mode is not supported")
521
522     def writeReg(self, addr, data, mode="HMC"):
523         '''Writes data to the specified register'''
524         if mode == "HMC":
525             '''Indicate a write cycle'''
526             GPIO.output(SEN, GPIO.HIGH)
527             GPIO.output(SDI, GPIO.LOW)
528             self.pulseSCK()
529
530             '''Send the 6 address bits'''
531             #print('Address')
532             self.shiftInData(addr, 6)
533
534             '''Send 24 data bits'''
535             #print('Data')
536             self.shiftInData(data, 24)
537             self.pulseSCK()
538
539             '''Finish the write cycle'''
540             time.sleep(0.001)
541             GPIO.output(SEN, GPIO.LOW)
542             time.sleep(0.001)
543             GPIO.output(SDI, GPIO.LOW)

```

```

543
544         else:
545             print("Writing in this mode is currently _
                    not supported")
546
547     def readReg(self, addr, mode="HMC"):
548         '''Reads data from the specified register'''
549         if mode == "HMC":
550             '''Indicate a read cycle'''
551             GPIO.output(SEN, GPIO.HIGH)
552             GPIO.output(SDI, GPIO.HIGH)
553             self.pulseSCK()
554
555             '''Send the 6 address bits'''
556             self.shiftInData(addr, 6)
557
558             '''Read the 24 data bits'''
559             recData = 0b000000000000000000000000
560             self.pulseSCK()
561             for i in np.arange(24):
562                 if GPIO.input(SDO):
563                     recData += 1<<(23-i)
564
565                 self.pulseSCK()
566
567             '''Finish the write cycle'''
568             time.sleep(0.001)
569             GPIO.output(SEN, GPIO.LOW)
570             time.sleep(0.001)
571             GPIO.output(SDI, GPIO.LOW)
572
573             return recData
574
575         else:
576             print("Reading in this mode is currently _
                    not supported")
577
578
579     def pulseSCK(self):
580         '''Send a rising edge pulse on the clock line'''
581         GPIO.output(SCK, GPIO.LOW)
582         GPIO.output(SCK, GPIO.HIGH)
583
584     def shiftInData(self, data, length):
585         for i in np.arange(length):
586             if data&(1<<((length-1)-i)) > 0:
587                 GPIO.output(SDI, GPIO.HIGH)
588             else:
589                 GPIO.output(SDI, GPIO.LOW)
590             self.pulseSCK()
591
592     def checkDeviceID(self):

```

```

593         '''Checks if the device ID is correct'''
594         print("Checking device ID..."),
595         id = self.readReg(0x00)
596         print(str(hex(id))+"..."),
597         if(id == 0xa7975):
598             print("[OK]")
599         else:
600             print("[FAIL]")
601
602
603     def LDLookup(self, window):
604         '''Returns the ideal speed and divide settings for
        the window size'''
605         window = window*1e9
606         if(window < 7.6):
607             divide = 0b000
608             if(window < 6.5):
609                 speed = 0b00
610             elif(window < 7):
611                 speed = 0b01
612             elif(window < 7.1):
613                 speed = 0b01
614             elif(window < 7.6):
615                 speed = 0b11
616         elif(window < 10.2):
617             divide = 0b001
618             if(window < 8):
619                 speed = 0b00
620             elif(window < 8.9):
621                 speed = 0b01
622             elif(window < 9.2):
623                 speed = 0b01
624             elif(window < 10.2):
625                 speed = 0b11
626         elif(window < 15.4):
627             divide = 0b010
628             if(window < 11):
629                 speed = 0b00
630             elif(window < 12.8):
631                 speed = 0b01
632             elif(window < 13.3):
633                 speed = 0b01
634             elif(window < 15.4):
635                 speed = 0b11
636         elif(window < 26):
637             divide = 0b011
638             if(window < 17):
639                 speed = 0b00
640             elif(window < 21):
641                 speed = 0b01
642             elif(window < 22):
643                 speed = 0b01

```

```

644         elif(window < 26):
645             speed = 0b11
646     elif(window < 47):
647         divide = 0b100
648         if(window < 29):
649             speed = 0b00
650         elif(window < 36):
651             speed = 0b01
652         elif(window < 38):
653             speed = 0b01
654         elif(window < 47):
655             speed = 0b11
656     elif(window < 88):
657         divide = 0b101
658         if(window < 53):
659             speed = 0b00
660         elif(window < 68):
661             speed = 0b01
662         elif(window < 72):
663             speed = 0b01
664         elif(window < 88):
665             speed = 0b11
666     elif(window < 172):
667         divide = 0b110
668         if(window < 100):
669             speed = 0b00
670         elif(window < 130):
671             speed = 0b01
672         elif(window < 138):
673             speed = 0b01
674         elif(window < 172):
675             speed = 0b11
676     elif(window < 338):
677         divide = 0b111
678         if(window < 195):
679             speed = 0b00
680         elif(window < 255):
681             speed = 0b01
682         elif(window < 272):
683             speed = 0b01
684         elif(window < 338):
685             speed = 0b11
686     else:
687         divide = 0b111
688         speed = 0b11
689     return divide , speed

```

Listing F.8: FPGA Control Python module for ARRTY host

```

691 import RPi.GPIO as GPIO
692 import time
693 import numpy as np
694

```


APPENDIX F. AERIAL REAL-TIME TRANSIENT ANALYSER
DEVELOPMENT

206

```

695 '''Pin_connections'''
696 CLK = 8
697 DIN = 25
698 DOUT = 11
699 SEN = 9
700
701 buffer = 200 # Adds a buffer to ensure the no of samples
702 overread = 300
703
704 class FPGACon():
705     def __init__(self):
706         '''Set_up_the_pins_to_the_digital_board'''
707         self.setupPins()
708         time.sleep(0.1)
709         GPIO.output(SEN, GPIO.HIGH)
710         time.sleep(0.1)
711         GPIO.output(SEN, GPIO.LOW)
712         time.sleep(0.1)
713
714         self.sampleLength = 1024
715
716     def setupPins(self):
717         '''Sets_up_the_GPIO_pins'''
718         '''Configure_the_behaviour_of_the_GPIO_pins'''
719         GPIO.setwarnings(False)
720         GPIO.setmode(GPIO.BCM)
721         GPIO.setup(CLK, GPIO.OUT)
722         GPIO.setup(DIN, GPIO.OUT)
723         GPIO.setup(DOUT, GPIO.IN)
724         GPIO.setup(SEN, GPIO.OUT)
725
726         '''Set_default_states'''
727         GPIO.output(CLK, GPIO.LOW)
728         GPIO.output(DIN, GPIO.LOW)
729         GPIO.output(SEN, GPIO.LOW)
730
731     def readData(self, customLength=-1):
732         '''Reads_all_of_the_data_from_the_fifo'''
733         if not customLength == -1:
734             length = customLength
735         else:
736             length = self.sampleLength
737
738         data = []
739         GPIO.output(DIN, GPIO.LOW)
740         GPIO.output(SEN, GPIO.HIGH)
741         for x in np.arange(length + buffer + overread):
742             byte = 0
743             for i in np.arange(8):
744                 self.pulseCLK()
745                 if (GPIO.input(DOUT) == 1):
746                     byte = byte|(0b1<<i)

```

```

747         if x < length + buffer/2 and x >= buffer
748             /2:
749             data.append(byte - 128)
750         return data
751
752     def setSampleLength(self, length):
753         '''Sets the no. of samples to store in the fifo
754         buffer
755         .....max=65000'''
756         '''Send set sample length command'''
757         self.sendCommand(0b00000000000010)
758         '''Send the sample length'''
759         self.sendCommand(length + buffer)
760         self.sampleLength = length
761
762     def setTriggerLevel(self, level):
763         '''Sets the trigger level'''
764         '''Send set trigger level command'''
765         self.sendCommand(0b00000000000001)
766         '''Send the trigger level'''
767         self.sendCommand(level)
768
769     def setTrigger(self):
770         '''Sets the device in trigger mode'''
771         '''Send set trigger command'''
772         self.sendCommand(0b00000000000011)
773
774     def readDOUT(self):
775         '''Returns the level of dout'''
776         return GPIO.input(DOUT)
777
778     def sendCommand(self, command):
779         '''Send a 16 bit command'''
780         GPIO.output(DIN, GPIO.LOW)
781         GPIO.output(CLK, GPIO.LOW)
782         GPIO.output(SEN, GPIO.HIGH)
783         for i in np.arange(16):
784             if command&(0b1<<i) > 0:
785                 GPIO.output(DIN, GPIO.HIGH)
786             else:
787                 GPIO.output(DIN, GPIO.LOW)
788             self.pulseCLK()
789         GPIO.output(SEN, GPIO.LOW)
790         GPIO.output(DIN, GPIO.LOW)
791         GPIO.output(CLK, GPIO.LOW)
792
793     def pulseCLK(self):
794         '''Pulse the clock pin'''
795         GPIO.output(CLK, GPIO.LOW)
796         GPIO.output(CLK, GPIO.HIGH)

```

Test Tone Measurements

With the full ARRTY device implemented some test tones were measured to test its performance. It should be noted that the signal integrity of the board was compromised when the high-frequency ground return layer was omitted during fabrication. As a result, unwanted coupling between the on-board systems has limited the performance of the device. Also, interleaving artefacts cause a significant amount of unwanted spurs. Nevertheless, it is possible to reduce these signals by calibrating the gain, offset and delay to each of the four interleaved amplifiers. Fig. F.15 to F.17 show the spectrum of a 60 us sample when applying a 90 MHz tone at different amplitude levels without calibrating the interleaved amplifiers.

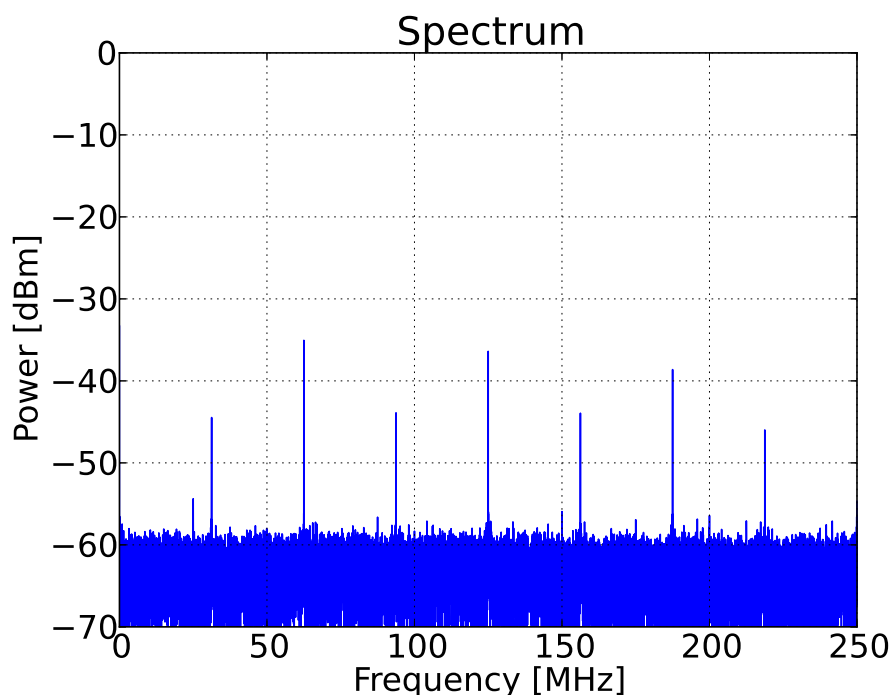


Figure F.15: Spectrum plot with $50\ \Omega$ load connected to the input.

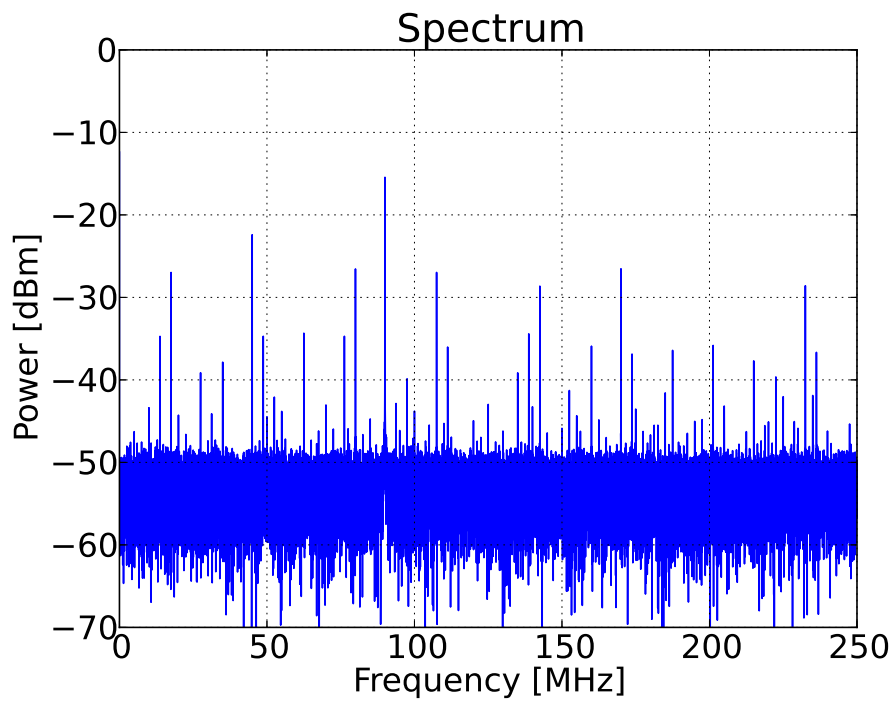


Figure F.16: -12.5 dBm 90 MHz tone applied at input.

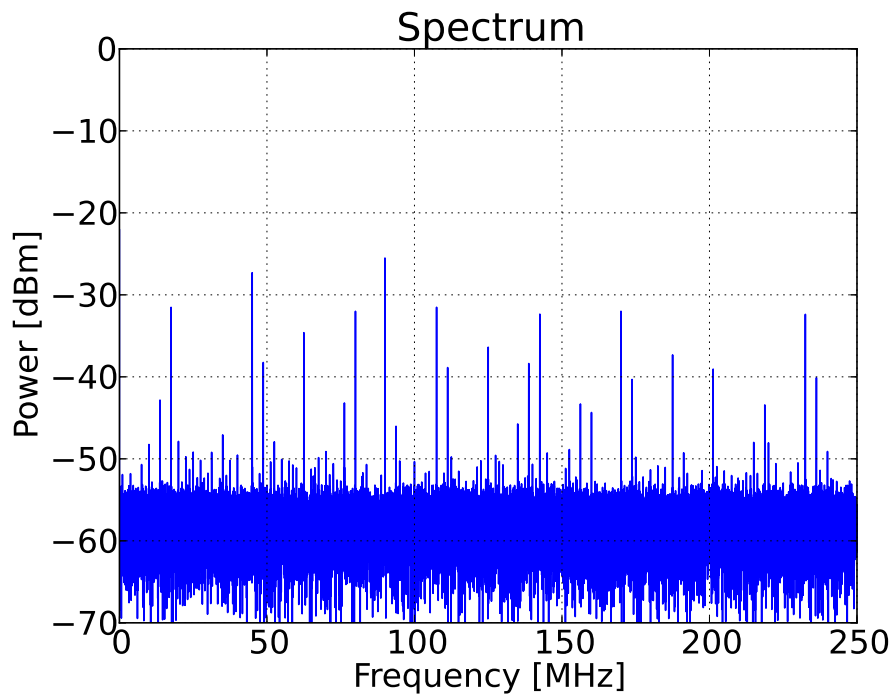


Figure F.17: -22.5 dBm 90 MHz tone applied at input.

Appendix G

Antenna Beam Scanning

In the previous section, it was shown how the Multi-copter could be used to scan the beam pattern of an antenna if the elements were aligned with the horizontal. Here an antenna beam scanning measurement is presented. The dual-antenna scheme was designed for RFI and propagation measurements. This is only a demonstration of the dual-antenna performance used for antenna characterisation.

The AUT, seen in Fig. G.1, is a multi-mode antenna. Its differential inputs allow it to act as a vertical monopole or horizontal dipole when driven by common or differential mode signals respectively. A rat-race coupler were used to drive the antenna in the required mode using a single generator port. The generator was set to a frequency of 960 MHz at 18 dBm. The Multi-



Figure G.1: Multi-mode antenna placed in the middle of a large sports field aligned East to West. In this case, the Rat-race coupler was used to drive the antenna in its horizontal dipole mode.

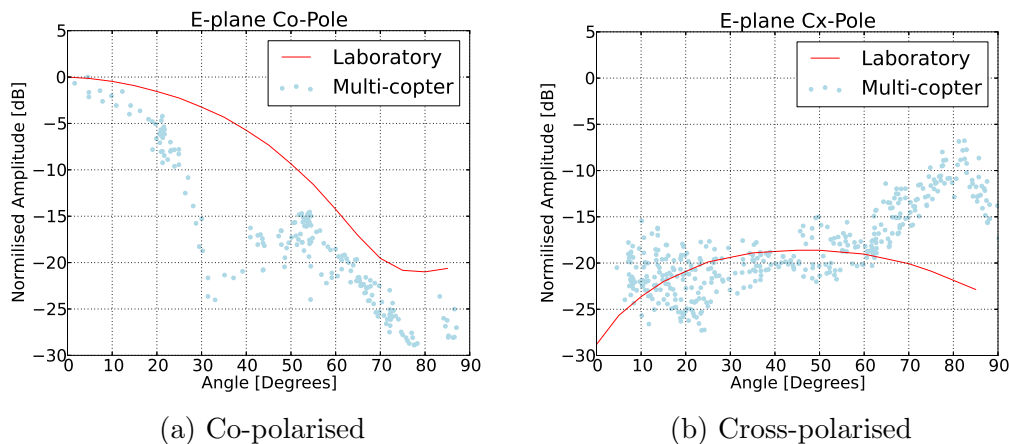


Figure G.2: E-plane dipole antenna measurement using Multi-copter compared to laboratory characterised data. Deviation in measurements are caused by the null region de-embedding of on-board antennas.

copter flight paths were aligned East-West and North-South over the zenith of the antenna. To de-embed the polarisation during the measurement the orientation of the Multi-copter was chosen so that antenna 1 and 2 were aligned North-South and East-West respectively.

The measurements compared to the lab verified performance of the antenna can be seen in Fig. G.2. The dual-antenna configuration was designed to give the Multi-copter a quasi-isotropic antenna pattern. This was achieved by selecting the appropriate receiving antenna given the current vehicle position and orientation to the source. However, this antenna selection scheme cannot be used when characterising an AUT. This causes measurements to pass through the null regions of the on-board antennas and result in an under or over compensation during the de-embedding process. The effect of this can be seen in the measurements where a large deviation is visible at around 30 degrees in Fig. G.2a which corresponds to the on-board, co-polarised antenna null region. The same mechanism is seen in Fig. G.2b where the de-embedding process over-estimates the on-board null pattern. In order to eliminate these errors, null regions need to be characterised more carefully. Also, it is possible to avoid these null regions all together by dynamically orientating the Multi-copter during the scan. However, this will be beyond the scope of this project. In the non-affected regions, the correlation between the field and laboratory measured data is good, and polarisation discrimination between the antennas are clear.

The aim of this section was to test the limitations of this dual-antenna configuration. It was found that the a high degree of accuracy is needed in the null regions of the on-board antennas. Without the latter, errors will be introduced during the de-embedding process when used in this configuration. Multi-copters with an appropriately aligned antenna (horizontal in this

case) would be much better suited towards antenna characterisation. The dual-antenna configuration used in this work is much better suited towards propagation measurements where the source is in most cases known.

Appendix H

Multicopter Measurements

Fig. H.1 to H.3 shows all of the berm diffraction measurements. As described in the text, these comparisons were used to calibrate the height for a single knife-edge diffractor. This diffractor was chosen as a simple approximation for the shielding of the berm.

Fig. H.4 shows the individually measured positions for the KAPB vertical-shielding flights. During these flights, a transmitter was placed in multiple rooms inside the building. This helped reverberate the measurement. At the time of these measurements, the KAPB was still in its final construction phases. Overall, these measurements show how shielding performance increases at higher frequencies. Additionally, shielding is in most cases lower at higher altitudes due to the structure of the building.

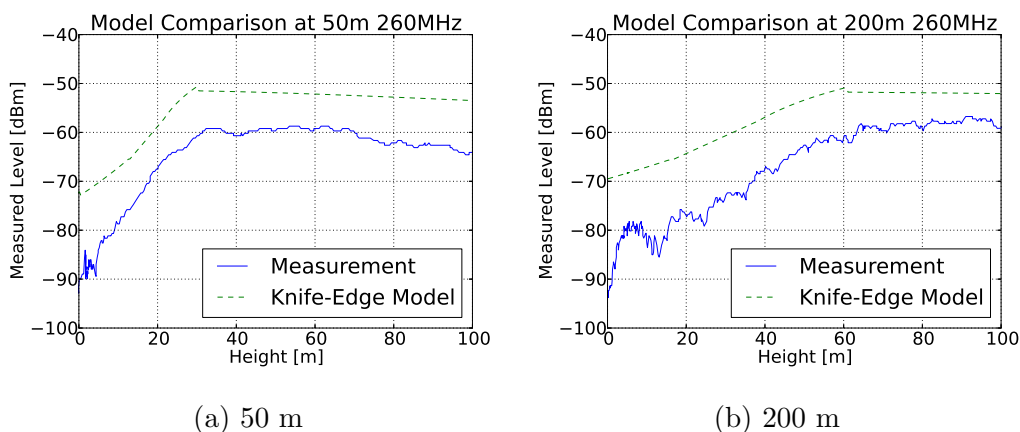


Figure H.1: Diffraction measurements at two discrete points behind berm at 260 MHz. These measurements are compared to the calibrated knife-edge model.

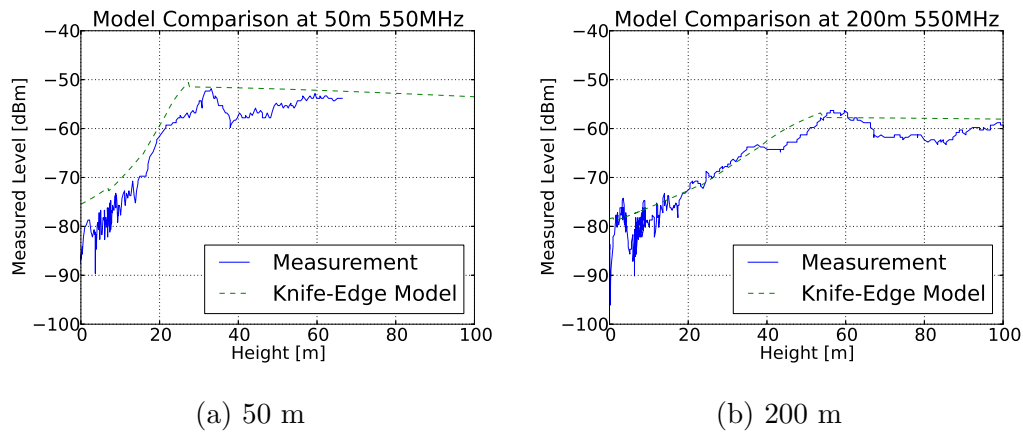


Figure H.2: Diffraction measurements at two discrete points behind berm at 550 MHz. These measurements are compared to the calibrated knife-edge model.

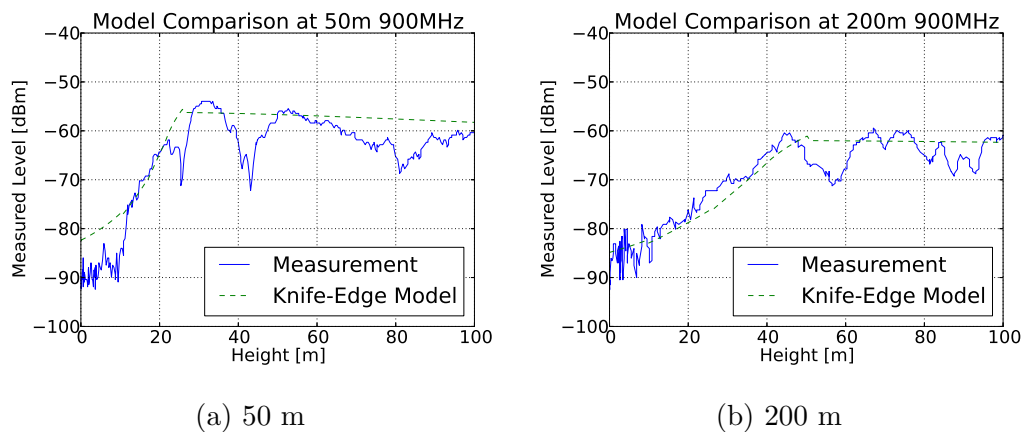
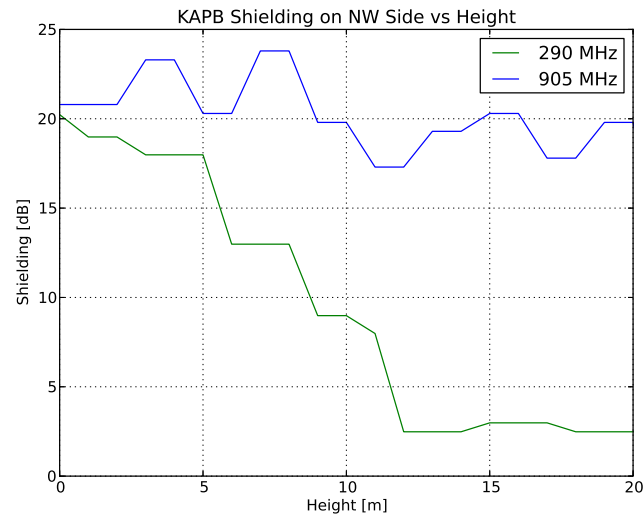
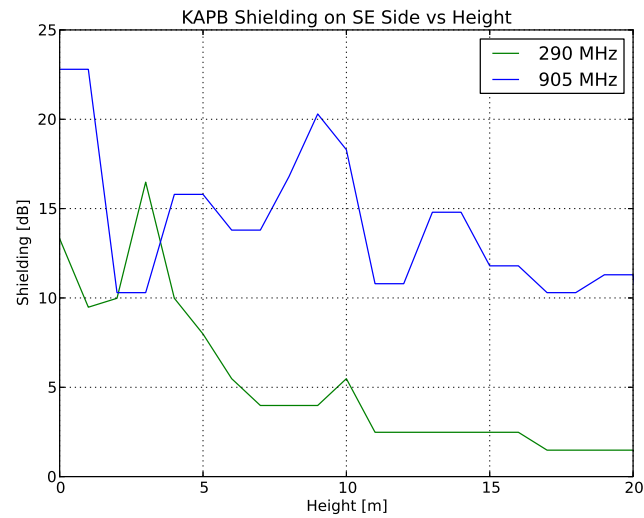


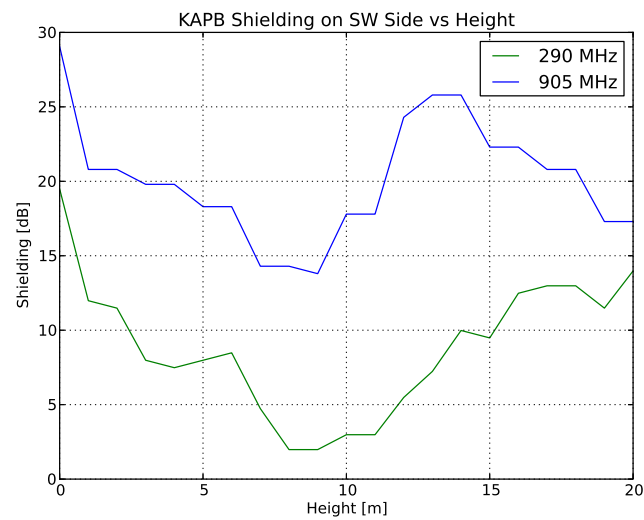
Figure H.3: Diffraction measurements at two discrete points behind berm at 900 MHz. These measurements are compared to the calibrated knife-edge model.



(a) NW side of building



(b) SE side of building



(c) SW side of building

Figure H.4: Vertical-shielding measurements for the remaining three sides of the building. The berm side has been discussed in the text.

Bibliography

- [1] ASU: MWA website. 2015. [Online; accessed 3-July-2015].
Available at: www.mwatelescope.org
- [2] CSIRO: ASKAP website. 2015. [Online; accessed 2-July-2015].
Available at: www.atnf.csiro.au/projects/askap/index.html
- [3] SKA: SKA website. 2015. [Online; accessed 6-August-2015].
Available at: www.skatelescope.org
- [4] Otto, A. J. et al.: Complex Permittivity Measurements of Karoo Soil for the Square Kilometre Array. *Proceedings of the 13th International Conference on Electromagnetics in Advanced Applications, Torino, Italy*, Sept 2011.
- [5] Groch, M. : *HV Transmission Line and Tower Inspection Safe-Fly Zone Modelling and Metrology*. Master's thesis, Stellenbosch University, Dec 2013.
- [6] Kibet-Langat, P. : *Power-line Sparking Noise Characterisation in the SKA Environment*. Ph.D. thesis, Dept. Elect. Eng., Stellenbosch University, 2011.
- [7] du Plessis, J. : Evaluation of Terrestrial Signal Propagation Characteristics for MeerKAT(SKA): Soil Properties. Dec 2011.
- [8] Matthysen, N. : *Time Domain Metrology for MeerKAT systems*. Master's thesis, Stellenbosch University, Dec 2014.
- [9] SKASA: SKA Africa website. 2015. [Online; accessed 4-August-2015].
Available at: www.ska.ac.za
- [10] ALMA: ALMA Website. 2014. [Online; accessed 6-August-2014].
Available at: www.almaobservatory.org
- [11] NRAO: VLA website. 2014. [Online; accessed 3-September-2014].
Available at: www.vla.nrao.edu
- [12] Regulations on Radio Astronomy Protection Levels in Astronomy Advantage Areas Declared for the Purposes of Radio Astronomy. 2007.
- [13] Botha, A. R. : *Development of a Real-Time Transient Analyser for the SKA*. Master's thesis, Stellenbosch University, March 2014.

- [14] Davidson, D. B. : *Computational Electromagnetics for RF and Microwave Engineering*. 2nd edn. Cambridge University Press, 2011.
- [15] Harrington, R. F. : *Field Computation by Moment Methods*. IEEE Press, 1993.
- [16] Pozar, D. M.: *Microwave Engineering*. 3rd edn. John Wiley & Sons, Inc., 2005.
- [17] FEKO: FEKO website. 2015. [Online; accessed 20-August-2015].
Available at: www.feko.info
- [18] Trevor Marshall: NEC2 unofficial website. 2015. [Online; accessed 3-July-2015].
Available at: www.nec2.org
- [19] Remcom: XFDTD website. 2015. [Online; accessed 3-July-2015].
Available at: <http://www.remcom.com/xf7>
- [20] CST: CST website. 2015. [Online; accessed 3-July-2015].
Available at: <https://www.cst.com/>
- [21] James, G. L. : *Geometrical Theory of Diffraction for Electromagnetic Waves*. 3rd edn. Peter Peregrinus Ltd., 1986.
- [22] Seybold, J. S.: *Introduction to RF Propagation*. 1st edn. John Wiley & Sons, Inc., 2005.
- [23] Haslett, C.: *Essentials of Radio Wave Propagation*. 1st edn. Cambridge University Press, 2008.
- [24] Kerr, D. E.: *Propagation of Short Radio Waves*. Dover Publications, Inc., New York, NY, 1951.
- [25] Schelleng, J. C. et al.: Ultra-Short-Wave Propagation. *Proceedings of the Institute of Radio Engineers*, vol. 21, pp. 427–463, March 1933.
- [26] Bullington, K.: Radio Propagation at Frequencies Above 30 Megacycles. *Proceedings of the IRE*, vol. 35, pp. 1122–1136, Oct 1947.
- [27] Bullington, K.: Radio Propagation Variations at VHF and UHF. *Proceedings of the IRE*, vol. 38, pp. 27–32, Jan 1950.
- [28] Bullington, K. et al.: Results of propagation test at 505 Mc and 4090 Mc on beyond horizon paths. *IRE Transactions on Communications*, vol. 4, pp. 104–111, March 1956.
- [29] Bullington, K.: Radio Transmission beyond the Horizon in the 40-to 4,000-MC Band. *Proceedings of the IRE*, vol. 41, pp. 132–135, Jan 1953.
- [30] Friis, H. T.: A Note on a Simple Transmission Formula. *Proceedings of the IRE*, vol. 34, pp. 254–256, May 1946.

- [31] Bullington, K.: Reflection Coefficients of Irregular Terrain. *Proceedings of the IRE*, vol. 42, pp. 1258–1262, Aug 1954.
- [32] Burrows, C. R. and Marion C. G.: The Effect of the Earth's Curvature on Ground-Wave Propagation. *Proceedings of the IRE*, vol. 29, pp. 16–24, Jan 1941.
- [33] Burrows, C. R.: Radio Propagation Over Plane Earth-Field Strength Curves. *Bell System Technical Journal*, vol. 16, pp. 45–75, Oct 1937.
- [34] Norton, K. A.: The Propagation of Radio Waves over the Surface of the Earth and in the Upper Atmosphere. *Proceedings of the Institute of Radio Engineers*, vol. 24, pp. 1367–1387, Oct 1936.
- [35] Robertson, S. D.: The Effect of Rain upon the Propagation of Waves in the 1- and 3-Centimeter Regions. *Proceedings of the IRE*, vol. 34, pp. 178–180, April 1946.
- [36] Gordon, W. E.: Radio Scattering in the Troposphere. *Proceedings of the IRE*, vol. 43, pp. 23–28, Jan 1955.
- [37] Bullington, K.: Characteristics of Beyond-the-Horizon Radio Transmission. *Proceedings of the IRE*, vol. 43, pp. 1175–1180, Oct 1955.
- [38] Dickson, F. H.: Large Reductions of VHF Transmission Loss and Fading by the Presence of a Mountain Obstacle in Beyond-Line-of-Sight Paths. *Proceedings of the IRE*, vol. 41, pp. 967–969, Aug 1953.
- [39] Egli, J. J.: Radio Propagation Above 40 MC Over Irregular Terrain. *Proceedings of the IRE*, vol. 45, no. 10, pp. 1383–1391, 1957.
- [40] Longley, A. G. and Rice, P. L.: Prediction of Tropospheric Radio Transmission Loss Over Irregular Terrain. A Computer Method. Tech. Rep., Institute for Telecommunication Sciences Boulder co., 1968.
Available at: <http://www.dtic.mil/dtic/tr/fulltext/u2/676874.pdf>
- [41] ITU-R P.1546-5. Tech. Rep., International Telecommunications Union, 2013.
- [42] Okumura, Y. et al.: Field Strength and its Variability in VHF and UHF land-mobile radio service. *Review of the Electrical Communication Laboratories*, vol. 16, no. 9-10, pp. 825–873, 1980.
- [43] Hata, M.: Empirical Formula for Propagation Loss in Land Mobile Radio Services. *IEEE Transactions on Vehicular Technologies*, vol. 29, no. 3, pp. 317–325, 1980.
- [44] Lee, W. C. Y.: *Mobile Cellular Telecommunication System, Analog & Digital*. 2nd edn. McGraw Hill, 1995.
- [45] Miller, D. A. B.: Huygens's Wave Propagation Principle Corrected. *Optics Letters*, vol. 16, pp. 1370–1372, Sept 1991.

- [46] Huygens, C.: *Traité de la Lumière*. Leyden, 1690.
- [47] LS-Telcom: LS of SA website. 2014. [Online; accessed 12-January-2014]. Available at: <http://www.lsofsa.co.za/>
- [48] Krause, M. : *Calibration of the LOFAR Antennas*. Master's thesis, Radboud University Nijmegen, 2013.
- [49] Virone, G. et al. : Antenna Pattern Measurements with a Flying Far-Field Source (Hexacopter). *Proceedings of the 2014 International Conference on Antenna Measurements and Applications (CAMA), Antibes, Juan-Les-Pins*, pp. 16–19, Nov 2014.
- [50] Virone, G. et al. : UAV-based Radiation Pattern Verification for a Small Low-Frequency Array. *Proceedings of the 2014 International Symposium on Antennas and Propagation (APSURSI), Memphis, TN*, pp. 6–11, July 2014.
- [51] Virone, G. et al. : Antenna Pattern Verification System Based on a Micro Unmanned Aerial Vehicle (UAV). *IEEE Antenna and Wireless Propagation Letters*, pp. 169–172, Feb 2014.
- [52] Balanis, C. A.: *Advanced Engineering Electromagnetics*. 2nd edn. John Wiley & Sons, Inc., 2012.
- [53] Venkatesh, M. S. and Raghavan, G. S. V.: An Overview of Dielectric Properties Measuring Techniques. *Canadian Biosystems Engineering*, vol. 47, no. 7, pp. 15–30, 2005.
- [54] Rossouw, D. J. : *Probes in HF Metrology*. Master's thesis, Stellenbosch University, 2010.
- [55] Nicolson, A. M. and Ross, G. F.: Measurement of The Intrinsic Properties of Materials by Time-Domain Techniques. *IEEE Transactions on Instrumentation and Measurement*, vol. 19, no. 4, pp. 377–382, 1970.
- [56] Weir, W. B.: Automatic Measurement of Complex Dielectric Constant and Permeability at Microwave Frequencies. *Proceedings of the IEEE*, vol. 62, no. 1, pp. 33–36, 1974.
- [57] Rimbi, M. : *Wood Microwave Dielectric Heating and Measurement of Material Properties*. Ph.D. thesis, Stellenbosch University, 2004.
- [58] Baker-Jarvis, J. and Vanzura, E. J. and Kissick, W. A.: Improved Technique for Determining Complex Permittivity with the Transmission/Reflection Method. *IEEE Transactions on Microwave Theory and Techniques*, vol. 38, no. 8, pp. 1096–1103, 1990.
- [59] van der Merwe, J. : *Uniform Multiconductor Transmission Lines Above a Dissipative Earth: Direct FDTD Analysis and Experimental Validation*. Ph.D. thesis, University of Stellenbosch, 1998.

- [60] Luukkonen, O. et al.: A Stepwise Nicolson-Ross-Weir-Based Material Extraction Method. *IEEE Antennas and Wireless Propagation Letters*, vol. 10, pp. 1295–1298, 2011.
- [61] Geyer, R. G. : Dielectric Characterization and Reference Materials. Tech. Rep. 1338, National Institute of Standards, 1990.
- [62] Baker-Jarvis, J. et al.: Dielectric & Magnetic Properties of Printed Wiring Boards & Other Substrate Materials. Tech. Rep. 1512, National Institute of Standards, 1999.
- [63] Smith, A. G.: *An Experimental Study of Artificial Isotropic Chiral Media at Microwave Frequencies*. Ph.D. thesis, Stellenbosch University, 1994.
- [64] Kruppa, W. and Sodomsy, K. F.: An Explicit Solution for the Scattering Parameters of a Linear Two-Port Measured with an Imperfect Test Set. *IEEE Transactions on Microwave Theory and Techniques*, pp. 122–123, 1971.
- [65] Arslanagic, S. et al.: A Review of the Scattering Parameter Extraction Method with Clarification of Ambiguity Issues in Relation to Metamaterial Homogenization. *IEEE Antennas and Propagation Magazine* 55, vol. 55, no. 2, pp. 91–106, 2013.
- [66] Keller, J. B.: Geometrical Theory of Diffraction. *Journal of the Optical Society of America*, vol. 52, no. 2, pp. 116–130, 1962.
- [67] Kouyoumjian, R. G. and Pathak, P. H.: A Uniform Geometrical Theory of Diffraction for an Edge in a Perfectly Conducting Surface. *Proceedings of the IEEE*, vol. 62, no. 11, pp. 1448–1461, 1974.
- [68] Lee, T-H. and Burnside, W. D.: Performance Trade-off Between Serrated Edge and Blended Rolled Edge Compact Range Reflectors. *IEEE Transactions on Antennas Propagation*, vol. 44, no. 1, pp. 87–96, 1996.
- [69] Personal correspondence with Danie le Roux from EMSS/Altair.
- [70] Jacobs, U. et al.: Extended Solution Methods in FEKO Suite 5.5: Ray- Launching GO, FEM Modal Ports, Periodic Boundary Conditions, and Windscreen Antenna Modeling. *Proceedings of the 26th Annual Review of Progress in Applied Computational Electromagnetics, Tampere, Finland*, April 2010.
- [71] Pienaar, H. et al.: Karoo Array Telescope Berm Shielding: Efficient Computational Modeling and Multicopter Measurement. *IEEE Transactions on Electromagnetic Capability*. Under Review.
- [72] Collin, R. E.: *Antennas and Radiowave Propagation*. McGraw-Hill, 1985.
- [73] Ilgner, R. G.: *A Comparative Analysis of the Performance and Deployment Overhead of Parallelized Finite Difference Time Domain (FDTD) Algorithms on a Selection of High Performance Multiprocessor Computing Systems*. Ph.D. thesis, Stellenbosch University, Dec 2013.

- [74] 3DR: 3D Robotics website. 2015. [Online; accessed 5-March-2015].
Available at: www.3drobotics.com
- [75] Maloney, J. G. and Smith, G. S. : A Study of Transient Radiation from the Wu-King Resistive Monopole - FDTD Analysis and Experimental Measurements. *IEEE Transactions on Antennas and Propagation*, vol. 41, no. 5, pp. 668–676, 1993.
- [76] Wu, T. T. and King, R. W. P. : The Cylindrical Antenna with Nonreflecting Resistive Loading . *IEEE Transactions on Antennas and Propagation*, vol. 13, pp. 369–373, 1965.
- [77] Wu, T. T. and King, R. W. P.: Correction: The cylindrical antenna with nonreflecting resistive loading. *IEEE Trans. Antennas Propagat.*, vol. 13, p. 998, November 1965.
- [78] Parini, C. et al.: *Theory and Practice of Modern Antenna Range Measurements*. IET, London, 2015.
- [79] Weisstein, E. W.: "Rotation Matrix." From MathWorld. 2014. [Online; accessed 13-July-2014].
Available at: mathworld.wolfram.com/RotationMatrix.html
- [80] Regulations on Radio Astronomy Protection Levels in Astronomy Advantage Areas Declared for the Purposes of Radio Astronomy. 10 February 2012.
- [81] ITU-R P.526-12. Tech. Rep., International Telecommunications Union, 2012.
- [82] MESA Solutions: MESA Solutions Website. 2015. [Online; accessed 23-October-2015].
Available at: <http://www.mesasolutions.co.za/>
- [83] ITU-R P.452-15. Tech. Rep., International Telecommunications Union, 2013.
- [84] Lee, W. C. Y.: *Mobile Communication Engineering, Theory and Applications*. 2nd edn. McGraw-Hill, New York, 1998.
- [85] Marais, J. I. F. : *A Permittivity Measurement System for High Frequency Laboratories*. Master's thesis, Stellenbosch University, 2006.
- [86] Rossouw, D. J. et al.: Flanged Coaxial Material Probes: Characterizing Exterior CM Energy. *Microwave Measurement Symposium*, pp. 1–6, 2009.
- [87] KiCad: KiCad EDA Software Suite website. 2014. [Online; accessed 1-February-2014].
Available at: www.kicad-pcb.org
- [88] OSH-Park: OSH-Park website. 2013. [Online; accessed 12-July-2013].
Available at: www.oshpark.com
- [89] T-Motor: RC Tiger Motors website. 2014. [Online; accessed 12-February-2014].
Available at: www.rctigermotor.com